The application of nonlinear programming on ration formulation for dairy cattle

J. Li,1 E. Kebreab,1 Fengqi You,2 J. G. Fadel,1 T. L. Hansen,3 C. VanKerkhove,4 and K. F. Reed3*
1Department of Animal Science, University of California, Davis 95616
2Robert Frederick Smith School of Chemical and Biomolecular Engineering, Cornell University, Ithaca, NY 14853
3Department of Animal Science, Cornell University, Ithaca, NY 14853
4School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853

ABSTRACT

The objective of this study was to compare the application of iterative linear programming (iteLP), sequential quadratic programming (SQP), and mixed-integer nonlinear programming-based deterministic global optimization (MINLP_DGO) on ration formulation for dairy cattle based on Nutrient Requirements of Dairy Cattle (NRC, 2001). Least-cost diets were formulated for lactating cows, dry cows, and heifers. Nutrient requirements including energy, protein, and minerals, along with other limitations on dry matter intake, neutral detergent fiber, and fat were considered as constraints. Five hundred simulations were conducted, with each simulation randomly selecting 3 roughages and 5 concentrates from the feed table in NRC (2001) as the feed resource for each of 3 animal groups. Among the 500 simulations for lactating cows, 57, 45, and 21 simulations did not yield a feasible solution when using iteLP, SQP, and MINLP_DGO, respectively. All the simulations for dry cows and heifers were feasible when using SQP and MINLP_DGO, but 49 and 11 infeasible simulations occurred when using iteLP for dry cows and heifers, respectively. The average ration costs per animal per day of the feasible solutions obtained by iteLP, SQP, and MINLP_DGO were $4.78 (±0.71), $4.45 (±0.65), and $4.44 (±0.65) for lactating cows; $2.39 (±0.52), $1.48 (±0.26), and $1.48 (±0.26) for dry cows; and $0.98 (±0.72), $0.97 (±0.15), and $0.91 (±0.14) for heifers, respectively. The average computation time of iteLP, SQP, and MINLP_DGO were 0.59 (±1.87) s, 1.15 (±0.62) s, and 58.69 (±68.45) s for lactating cows; 0.041 (±0.070) s, 0.76 (±0.37) s, and 14.84 (±39.09) s for dry cows; and 1.60 (±2.90) s, 0.51 (±0.19) s, and 16.45 (±45.56) s for heifers, respectively. In conclusion, iteLP had limited capability of formulating least-cost diets when nonlinearity existed in the constraints. Both SQP and MINLP_DGO handled the nonlinear constraints well, with SQP being faster, whereas MINLP_DGO was able to return a feasible solution under some situations where SQP could not.

Key words: feed formulation, nonlinear programming, dairy cattle, NRC (2001)

INTRODUCTION

Feed costs account for around 50 to 70% of the expenses of operating a dairy farm (Bozic et al., 2012), which highlights the importance of minimizing this cost when feasible. Diet formulation relies on the nutrient requirements of the animal, nutrient composition of available feeds, and nutrient interactions, all of which require systems for estimation, such as Nutrient Requirements of Dairy Cattle (NRC, 2001) and the Cornell Net Carbohydrates and Protein System (Fox et al., 2004). One method to formulate least-cost diets that fulfill an animal’s nutrient requirements is linear programming (LP), which optimizes a linear objective function subject to a set of linear constraints (Chandler and Walker, 1972; O’Connor et al., 1989). A linear objective function (or a linear constraint) is a function with a polynomial of degree equal to 1 or 0 (e.g., a + b1x1 + b2x2 + . . . + bnxn) so that the value to be optimized has a linear relationship with the variables. Several studies have implemented LP for ration formulation to explore the effects of feed composition variability in the constraints (Tozer, 2000), feed price variability in the objective function (Alqaisi and Schlecht, 2021), and objective functions other than least cost (Qu, et al., 2019). One limitation of LP is that it allows only linear objective functions and constraints, while some equations in the dairy nutrition model are nonlinear when adapted to an LP structure. Nonlinear functions include a broad range of functions, in which the change of the output is not proportional to the change of the input [e.g., x^2, exp(x), log(x), and piecewise function]. In NRC (2001), nonlinearity mainly exists in the cal-
culation of energy and protein contents of feeds. For energy, the TDN values of feeds are corrected by the animal intake level, calculated as follows:

\[
\text{IntakeLevel} = \begin{cases} 
1, & \text{if } \text{TotalTDN} < 0.35 \times 0.75 \\
\frac{\text{TotalTDN}}{0.35 \times 0.75}, & \text{if } \text{TotalTDN} \geq 0.35 \times 0.75 
\end{cases}
\]

where IntakeLevel (dimensionless) is the incremental intake above maintenance; TotalTDN (%) is dietary TDN concentration. Intake level is used to adjust the TDN value of feeds, because feed intake above maintenance would decrease the nutrient digestibility (NRC, 2001), which consequently affects metabolizable and net energy values. For protein, the calculation of passage rates for concentrate requires DMI and dietary concentrate percentage:

\[
K_p = 2.904 + 1.375 \times \frac{\text{DMI}}{\text{BW}} \times 100 - 0.02 \times \text{PercentConc},
\]

where \(K_p\) = protein passage rate of feed i, \%/h; PercentConc = dietary concentrate percentage = the amount of concentrate (kg)/total amount of feed (kg) × 100%, percent of DM. These rates are then used to calculate rumen degradable and undegradable protein (RDP and RUP) contents. The intake level and dietary concentrate percentage are unknown before formulating the diet, so adapting these equations to an optimization programming creates the nonlinearity.

Not many studies have investigated how to handle nonlinearity in ration optimization, which is important because increasingly more nonlinear equations may appear with advances in dairy nutrition. Moraes et al. (2012) used an iterative linear programming (iteLP) method to deal with the nonlinearity in the NRC (2001) for ration formulation. However, iteLP has certain limitations, which will be discussed in this paper. A nonlinear programming optimization strategy, sequential quadratic programming (SQP; Boggs and Tolle, 1995), was employed in the Cornell-Penn-Miner (CPM) dairy model to formulate rations based on the Cornell Net Carbohydrates and Protein System (Boston et al., 2000), but the performance of using SQP based on NRC (2001) is unknown. Mixed-integer, nonlinear programming-based, deterministic global optimization (MINLP_DGO) is another method to solve nonlinear programming problems and is capable of solving a broader range of problems involving continuous, binary, and integer variables compared with SQP.

Mathematical modeling has been an important technique to evaluate production and environmental impacts of dairy systems (France and Kebreab, 2008). Whole-farm models such as the Integrated Farm Systems Model can provide holistic estimates of production and environmental outcomes in response to changes in weather and management (Rotz et al., 2016; Veltman et al., 2018) and require a method to predict feed use and delivery over the timespan of the model simulation. The Integrated Farm Systems Model currently uses an iteLP approach for ration formulation with requirements modified from a previous version of the NRC dairy nutrient requirement model (NRC, 1989), the NRC beef nutrient requirement model (NRC, 2000), and early versions of the Cornell Net Carbohydrates and Protein System model (Rotz et al., 1999, 2016). Limitations of the ration formulation method in the Integrated Farm Systems Model are among the factors that have led our group to develop a new whole-farm model, the Ruminant Farm Systems Model, to evaluate connections between dairy system components, including animal husbandry and feeding, manure management, field and crop management, and feed storage (Kebreab et al., 2019). The nonlinear programming framework for ration formulation developed in this study will automate simulation of feed use and production within the Ruminant Farm Systems Model and represent an advancement over extant whole-farm models (Kebreab et al., 2019). The objectives of this study are (1) to introduce a methodology to formulate least-cost rations using MINLP_DGO and (2) to compare the performance of iteLP, SQP, and MINLP_DGO on ration formulation based on NRC (2001).

**METHODS**

Least-cost diets were formulated to meet the nutrient requirements of dairy cattle in different life stages, with nutrient compositions of feeds as described by the NRC (2001). Ration formulation constraints were determined based on NRC (2001) and included the requirements for net energy for maintenance (NEₘ, Mcal/d), lactation (NEₗ, Mcal/d), and growth (NE₉, Mcal/d), and metabolizable protein (MP, g/d), calcium (g/d), and phosphorus (g/d). We introduced additional ration formulation constraints to guide formulation of diets that meet requirements for rumen function. These included a fat constraint of less than 7% of diet DM (NRC, 2001), an NDF constraint of greater than 25% and less than 40% of diet DM (NRC, 2001; Moraes et al., 2012), and a constraint of forage NDF greater than 19% of diet DM (NRC, 2001). Finally, DMI was limited to be less than or equal to the predicted DMI.
in NRC (2001), so that low-quality and low-cost feeds would not be overfed. We provide detailed information of all the constraints in Supplemental File S1 (Li, 2021). Three optimization strategies, iteLP, SQP, and MINLP_DGO, use different approaches to deal with the nonlinearity existing in the energy and protein constraints as will be described.

**Optimization Strategies**

**iteLP.** Formulating least-cost diets using LP can be written as follows:

\[
\text{Min } Z = \sum_{j=1}^{m} c_j x_j, \quad [3]
\]

subject to \( \sum_{j=1}^{m} a_{ij} x_j \geq b_i \) for \( i = 1, 2, \ldots, n, \)

where \( Z (\$) \) is the diet cost; \( c_j (\$/kg of DM) \) is the feed price of feed \( j \); \( x_j \) (kg of DM) is the amount of feed \( j \); \( a_{ij} \) is the coefficient for \( x_j \) in constraint \( i \); and \( b_i \) is the lower bound of constraint \( i \). For example, if constraint \( i \) represents NE\textsubscript{M} requirement, then \( b_i \) is the minimum NE\textsubscript{M} requirement (Mcal) and \( a_{ij} \) is the NRC (2001) predicted concentration of NE\textsubscript{M} in feed \( j \) (Mcal/kg of DM). The iteLP method uses the following procedures to deal with the nonlinearity existing in Eq. [1] and [2]. The feed ingredients (\( a_{ij} \)) in iteLP starts at certain initial values and then are updated according to the DMI at the solution. The iteration process is repeated until DMI and dietary concentrate percentage at the solution are relatively constant, to ensure that energy and protein requirements are met (Moraes et al., 2012). However, we found that iterating based on DMI and dietary concentrate percentage does not always result in a satisfactory solution, depending on the feed ingredients. During the iteration process, 2 solutions may have similar DMI and concentrate percentage but very different values of dietary TDN concentration and RDP intake, leading to a large discrepancy between nutrient requirement and nutrient supply at the final solution. Therefore, the iteration was based on intake level and microbial crude protein (MCP) production in this study. The algorithm stopped when the differences of intake level and MCP production between 2 iterations were both lower than 0.1%. To prevent infinite iterations when intake level or MCP does not converge, we set the maximum number of iterations to be 1,000.

**SQP.** Sequential quadratic programming allows nonlinear constraints, so the nutrient requirement constraints can be built according to NRC (2001) directly. The basic structure of an SQP problem is as follows:

\[
\text{Min } Z = \sum_{j=1}^{m} c_j x_j, \quad [4]
\]

subject to \( g_i (\mathbf{x}) \leq 0, \) for \( i = 1, 2, \ldots, n, \)

where \( \mathbf{x} \) is a vector of feed inclusion rates (\( x_j \) defined as above) and \( g_i (\mathbf{x}) \) is constraint \( i \), which must be twice continuously differentiable with respect to all \( x_j \) in \( \mathbf{x} \). However, several equations in NRC (2001) result in either nondifferentiable or discrete constraints. For example, Eq. [1] is not differentiable at the point at which TotalTDN is equal to 0.35 BW\(^{0.75}\) and IntakeLevel is equal to 1. In addition, the calculation of MCP production (kg) is discrete:

\[
\text{MCP} = \min(0.13\text{TDN}_{\text{intake}}, 0.85\text{RDP}_{\text{intake}}), \quad [5]
\]

where TDN intake (kg) is the discounted TDN intake; RDP intake (kg) is the RDP intake. These constraints present challenges to formulating diets with SQP, which we discuss in the following section. Additionally, the SQP algorithm converges to a local optimum (the optimum within a neighboring set of solutions), which could be far away from the global optimum (the optimum among all possible solutions) solution in some complicated nonlinear problems (Boggs and Tolle, 1995).

**MINLP_DGO.** Another optimization strategy, MINLP_DGO, allows the use of a mixture of continuous and binary decision variables. Note that mixed-integer nonlinear programming (MINLP) includes a wide range of nonlinear optimization problems containing continuous, binary, and integer variables. Solving an MINLP problem usually involves multiple algorithms and techniques (Kronqvist et al., 2019). Several solvers, including Couenne (Burer, 2009), BARON (Tawarmalani and Sahinidis, 2013), and Gurobi (Gurobi Optimization LLC, 2021), find deterministic and global solutions in MINLP problems. Formulating least-cost diets using MINLP can be written thus:

\[
\text{Min } Z = \sum_{j=1}^{m} c_j x_j, \quad [6]
\]

subject to \( g_i (\mathbf{x}, \mathbf{y}) \leq 0, \) for \( i = 1, 2, \ldots, n, \)

where \( \mathbf{y} \) is a binary variable vector. Using binary variables enables the conversion of Eq. [1] and [5] into several constraints. For Eq. [1], the intake level calculation can be written into 3 constraints:

\[
1 - M \times (1 - y_1) \leq \text{IntakeLevel} \leq 1 \\
+ M \times (1 - y_1), \quad [7]
\]
where $M$ is a large positive number (suppose $M = 100,000$) and $y_1$ is a binary variable. When $y_1$ is equal to 0, Eq. (7) is ineffective, because it models an intake level greater than a very small number and smaller than a very large number; Eq. (8) is equivalent to TotalTDN greater than 0.035 BW$^{0.75}$ because it models a TotalTDN greater than 0.035 BW$^{0.75}$ and smaller than a large number; Eq. (9) is equivalent to intake level equal to TotalTDN/0.035 BW$^{0.75}$ because it models an intake level greater or equal to TotalTDN/0.035 BW$^{0.75}$ and smaller than or equal to TotalTDN/0.035 BW$^{0.75}$. Similarly, when $y_1$ is equal to 1, Eq. (9) is ineffective; Eq. (8) is equivalent to TotalTDN smaller than 0.035 BW$^{0.75}$; Eq. (7) is equivalent to intake level equal to 1. In summary, Eq. (7) through (9) together are equivalent to Eq. (1). The same technique can be used for the conversion of Eq. (5):

$$0.035BW^{0.75} - M \times y_1 \leq \text{TotalTDN} \leq 0.035BW^{0.75}$$

$$+ M \times (1 - y_1), \quad [8]$$

$$\frac{\text{TotalTDN}}{0.035 \times BW^{0.75}} - M \times y_1 \leq \text{IntakeLevel} \leq \frac{\text{TotalTDN}}{0.035 \times BW^{0.75}} + M \times y_1, \quad [9]$$

Similarly, when $y_1$ is equal to 1, Eq. (9) is ineffective; Eq. (8) is equivalent to TotalTDN smaller than 0.035 BW$^{0.75}$; Eq. (7) is equivalent to intake level equal to 1. In summary, Eq. (7) through (9) together are equivalent to Eq. (1). The same technique can be used for the conversion of Eq. (5):

$$0.13TDN_{\text{intake}} - M \times y_2 \leq \text{MCP} \leq 0.13TDN_{\text{intake}}$$

$$+ M \times y_2, \quad [10]$$

$$0.85RDP_{\text{intake}} - M \times (1 - y_2) \leq \text{MCP} \leq 0.85RDP_{\text{intake}}$$

$$+ M \times (1 - y_2), \quad [11]$$

$$0.13TDN_{\text{intake}} - M \times y_2 \leq 0.85RDP_{\text{intake}} \leq 0.13TDN_{\text{intake}} + M \times (1 - y_2), \quad [12]$$

where $y_2$ is a binary variable. Similar to the behavior of Eq. (7) through (9), when 0.13 TDN$_{\text{intake}}$ ≥ 0.85 RDP$_{\text{intake}}$, $y_2 = 1$, MCP = 0.85 RDP$_{\text{intake}}$; when 0.13 TDN$_{\text{intake}}$ < 0.85 RDP$_{\text{intake}}$, $y_2 = 0$, MCP = 0.13 TDN$_{\text{intake}}$.

**Simulation**

A simulation study was conducted to compare the 3 optimization strategies. We simulated 3 animal groups: lactating cows, dry cows, and heifers. The animal inputs needed to define the nutrient requirement constraints are summarized in Table 1. A total of 500 simulations were conducted by randomly selecting a set of feeds containing (arbitrarily) 5 concentrates and 3 forages from 100 feeds (feeds and prices shown in Supplemental File S2; Li, 2021) in the feed table provided by NRC (2001). Calcium phosphate (monobasic) was kept for all 500 sets of feeds, to minimize excess mineral feeding. The 2020 annual average feed prices were taken from the Pennsylvania State University feed price list (https://extension.psu.edu/files/feed-price-lists; State University Cooperative Extension, 2020). For each animal group, 3 least-cost diets were formulated based on each set of the randomly selected feeds using iteLP, SQP, and MINLP_DGO, respectively, which gave 4,500 (3 × 3 × 500) diets in total.

**Table 1.** Animal attributes of the lactating cows, dry cows and heifers for the simulation study

| Item                  | Lactating cows | Dry cows | Heifers |
|-----------------------|----------------|----------|---------|
| BW, kg                | 650            | 720      | 300     |
| Parity                | 2              | 3        | 0       |
| Milk production, kg   | 35             | —        | —       |
| Milk protein, %       | 3.0            | —        | —       |
| Milk fat, %           | 3.5            | —        | —       |
| Days in milk, d       | 50             | —        | —       |
| Days of pregnancy, d  | 0              | 260      | 0       |
| Age, mo               | —              | 9.5      |         |
| Calving interval, d   | 370            | 370      | 370     |

**Table 2.** Chemical compositions (% of DM) and price ($/kg of DM) of an example feed set

| Feed                  | CP    | EE$^1$ | NDF | ADF | Ca    | P    | TDN$^2$ | Price$^3$ |
|-----------------------|-------|--------|-----|-----|-------|------|---------|-----------|
| Almond hulls          | 6.5   | 2.9    | 36.8| 28.7| 0.28  | 0.13 | 58.4    | 0.16      |
| Canola meal           | 37.8  | 5.4    | 29.8| 20.5| 0.75  | 1.10 | 69.9    | 0.31      |
| Steam-flaked corn     | 9.4   | 4.2    | 9.5 | 3.4 | 0.04  | 0.30 | 91.7    | 0.18      |
| Corn silage           | 8.8   | 3.2    | 45.0| 28.1| 0.28  | 0.26 | 68.8    | 0.18      |
| Cotton seed           | 23.5  | 19.3   | 50.3| 40.1| 0.17  | 0.60 | 77.2    | 0.26      |
| Grass hay             | 13.3  | 2.5    | 57.7| 36.9| 0.66  | 0.29 | 59.7    | 0.24      |
| Legume hay            | 20.2  | 2.1    | 39.6| 31.2| 1.52  | 0.26 | 59.0    | 0.31      |
| Soybean meal          | 46.3  | 8.1    | 21.7| 10.4| 0.36  | 0.66 | 88.5    | 0.37      |
| Calcium phosphate monobasic | 0    | 0      | 0   | 0   | 16.40 | 21.60 | 0       | 0.96      |

$^1$EE = ether extract.

$^2$TDN are the standard values from the NRC (2001) table.

$^3$Feed prices are from Pennsylvania State University feed price list (State University Cooperative Extension, 2020).
**Evaluation**

To demonstrate the validity of each optimizer, all the formulated diets were evaluated with the NRC (2001) equations to examine the feed nutrient supply. The difference between requirement and supply was calculated for all the nutrient constraints. In addition, we compared the diet cost and computation time for 3 optimizers implemented with an i7-5500U computer processor (2.40 GHz). We report the results of diet cost and computation time as mean (±SD).

**Software**

All simulations and computations were conducted in Python 3.7.4 (Van Rossum and Drake, 2009). We used the SciPy package (Virtanen et al., 2020) as the solvers for iteLP and SQP. For MINLP_DGO, Gurobi (Gurobi Optimization LLC, 2021) was used because it is free to academic users and can be easily implemented in Python.

**RESULTS AND DISCUSSION**

**An Example Simulation**

A set of feeds in one simulation is shown in Table 2. The diets formulated based on the feeds by iteLP, SQP, and MINLP_DGO are shown in Table 3. For lactating cows and dry cows, the rations obtained by SQP and MINLP_DGO were the same, but those obtained by iteLP were more expensive. For heifers, the rations obtained by all 3 optimizers were very close.

**Feasibility and Nutrient Balance**

In formulating diets for lactating cows, 57 simulations for iteLP were infeasible, 6 of which were infeasible because the SciPy package could not find a feasible solution in the first several iterations. The other 51 simulations were infeasible because the maximum number of iterations was reached before the convergence of intake level and MCP. In these cases, the value of intake level or MCP oscillated between 2 or several values and failed to converge during the iteration process (Figure 1), which highlights a flaw of iteLP. For MINLP_DGO, 21 simulations had no feasible solutions. To investigate the reasons for infeasibility, the 21 simulations were rerun after adding 2 dummy feed variables, including a “super protein” feed with 100% of protein and a “super energy” feed with 30 Mcal of digestible energy. Both feeds were set to be very expensive (i.e., $100/kg of DM), so that they would not be used when other feeds can fulfill the protein and energy requirements. Their...
appearance in the solution indicates a lack of protein or energy from feeds. The results showed that all 21 simulations were infeasible due to protein deficiency. The same 21 simulations were also infeasible for SQP, and 24 additional simulations were infeasible due to failure to fulfill the protein requirement. These 24 simulations were feasible when using MINLP_DGO, which means SQP did not fully explore the feed potential in these cases. Optimization with SQP relies on the gradient of the objective function and constraints, but the MP constraint is not continuous due to the discrete choice between prediction of MCP based on energy or protein availability represented in Eq. [5]. The SciPy package calculates the gradient numerically instead of analytically, which makes fitting a discrete constraint into SQP possible. In the MP constraint, the MCP production is limited by either TDN intake or RDP intake, which creates 2 discrete value domains. In most cases, the optimum exists in one of the domains and is far away from the other, so the SQP solver is able to search for the optimum within one domain without influence from discreteness. However, sometimes when the optimum is close to the boundaries of the 2 domains, SQP may not be able to find a proper solution based on the gradient, due to the discreteness on the optimum direction. However, by including binary variables, MINLP_DGO is able to find the optimum through the branch-and-bound approach (Land and Doig, 1960), which is able to evaluate the suboptimal solutions in each domain.

Figure 1. An example of microbial crude protein (MCP) production oscillating between several values and failing to converge when using iterative linear programming (iteLP) to design feed formulation for lactating cows (using corn grain, cotton seed meal, corn distillers, corn cob, canola meal, mixed grass-legume hay, alfalfa meal, mixed grass-legume silage, and calcium phosphate monobasic).

Figure 2. Net energy of lactation (NEL) balance (diet NEL supply − animal NEL requirement), MP balance (diet MP supply − animal MP requirement), diet cost, and computation time of 500 simulations for lactating cows based on 3 optimizers. iteLP = iterative linear programming (443 feasible simulations), SQP = sequential quadratic programming (455 feasible simulations), MINLP_DGO = mixed-integer nonlinear programming-based deterministic global optimization (479 feasible simulations). The upper and lower edges of boxes represent the upper and lower quartiles of the data, respectively; the midline represents the median; the upper and lower whiskers measure the distance of 1.5 times the interquartile range (interquartile range = upper quartile − lower quartile) above and below the upper and lower quartiles, respectively; all the dots represent the outliers falling outside the whiskers.
and find the best one (Taylor, 2009). Heuristically, MINLP_DGO will consider both TDN-limited and RDP-limited cases for MCP production and select the better solution. The requirement of NE_L did not result in any infeasible solutions for SQP, even though the NE_L constraint contained a nondifferentiable function (Eq. [1]). Because IntakeLevel represents the incremental intake above maintenance, the IntakeLevel value of diets fulfilling the requirements of NE_M, NE_L, and NE_G at the same time would always be greater than 1, and the nondifferentiable point (IntakeLevel = 1) would not disrupt the solution. All the infeasible simulations for SQP and MINLP_DGO were also infeasible for iteLP. We also investigate the feasibility of different lactating groups by changing the milk production level (results not shown). The comparison among the 3 optimization strategies was similar, but more infeasible simulations were observed for higher milk production groups, who demand more protein, so it is more difficult to obtain a solution with the same feeds. When formulating diets for dry cows and heifers, a feasible solution was obtained for all the simulations using SQP and MINLP_DGO, but 49 and 11 infeasible simulations, respectively, occurred using iteLP. For dry cows and heifers, the protein requirement did not cause infeasibility in SQP, probably due to much smaller nutrient requirements of dry cows and heifers. All the infeasible simulations were removed for the following analysis.

Boxplots of the difference between requirements and estimated nutrient delivery of the ration solutions for lactating cows from all feasible simulations are shown in Figure 2. Only NE_L and MP balance values are shown, because the other constraints were linear and did not differ greatly between the 3 optimizers. The NE_L and MP balance values were positive for the solutions obtained by iteLP, SQP, and MINLP_DGO in all feasible simulations, but the NE_L balance values were higher for iteLP, which suggests a tendency of overfeeding energy for rations formulated with iteLP. Differences between the rations simulated for dry cows and heifers (Figure 3 and 4) by iteLP, SQP, and MINLP_DGO were similar to those for lactating cows, except that iteLP also showed a tendency of overfeeding protein.

**Ration Cost**

Average ration costs per animal obtained by iteLP, SQP, and MINLP_DGO were $4.78 (±0.71), $4.45

![Figure 3](image-url)

*Figure 3. Net energy of lactation (NE_L) balance (diet NE_L supply − animal NE_L requirement), MP balance (diet MP supply − animal MP requirement), diet cost, and computation time of 500 simulations for dry cows. iteLP = iterative linear programming (451 feasible simulations), SQP = sequential quadratic programming (500 feasible simulations), MINLP_DGO = mixed-integer nonlinear programming-based deterministic global optimization (500 feasible simulations). The upper and lower edges of boxes represent the upper and lower quartiles of the data, respectively; the midline represents the median; the upper and lower whiskers measure the distance of 1.5 times the interquartile range (interquartile range = upper quartile − lower quartile) above and below the upper and lower quartiles, respectively; all the dots represent the outliers falling outside the whiskers.*
Journal of Dairy Science Vol. 105 No. 3, 2022

(±0.65), and $4.44 (±0.65) for lactating cows; $2.39 (±0.52), $1.48 (±0.26), and $1.48 (±0.26) for dry cows; and $0.98 (±0.72), $0.97 (±0.15), and $0.91 (±0.14) for heifers, respectively (Figure 2, 3, and 4). Ration costs were similar for SQP and MINLP_DGO, which indicates that the solution obtained through SQP was very close to the global optimum in ration optimization. However, the costs obtained by iteLP were greater, especially for dry cows. The iteration process in iteLP does not push the solution along the optimum direction, but blindly replaces the initial value with the solution from last iteration; therefore, the ration cost is minimized only within each iteration instead of the whole process and is likely to be greater than the one obtained by SQP or MINLP_DGO.

The least-cost diets for a given level of milk production estimated by the algorithms we present here have practical applications for the dairy industry and can be implemented in whole-farm simulation studies. However, other methods of diet optimization, such as maximization of income over feed costs or income minus feed costs, are important to consider as well. In our study, milk production serves as an input and thus is not estimated in the optimization process. To use maximization of milk income minus feed cost as the objective function, instead of minimization of feed cost, we would need to consider the interactions among milk production, milk composition, ration formulation, and DMI, the complexity of which requires future studies to investigate.

**Computation Time**

The average computation time of iteLP, SQP, and MINLP_DGO were 0.59 (±1.87) s, 1.15 (±0.62) s, and 58.69 (±68.45) s for lactating cows; 0.041 (±0.070) s, 0.76 (±0.37) s, and 14.84 (±39.09) s for dry cows; and 1.60 (±2.90) s, 0.51 (±0.19) s, and 16.45 (±45.56) s for heifers, respectively (Figure 2, 3, and 4). The computation time of iteLP and SQP was much shorter than that of MINLP_DGO, especially when formulating diets for lactation cows. Because MINLP_DGO solves the problem involving binary variables, the resulting problem becomes computationally expensive. When the solver Gurobi solves MIPMLP problems, the best bound and the suboptimal solution are updated throughout the computation. The best bound is the upper bound of the objective function value for a maximization problem.
(or the lower bound of a minimization problem), and the suboptimal solution is the best feasible solution found so far. By default, the computation stops when the gap between the 2 values decreases below 0.01%, but it may take hours to obtain such a solution. The computation time was limited to 3 min in this study. In total, 98, 21, and 33 simulations for lactating cows, dry cows, and heifers, respectively, exceeded the time limit (computation time of 3 min), but the gaps for them were all less than 0.1%, meaning the solution found was close to the true optimum.

CONCLUSIONS

The linear optimization iteLP had limited capability to formulate least-cost diets when nonlinearity existed in the constraints. Both SQP and MINLP_DGO were able to handle nonlinear constraints well, with SQP being faster, but MINLP_DGO was able to return a feasible solution under some situations where SQP could not. Thus, both nonlinear programming frameworks for least-cost ration formulation represent advancement over previous linear programming techniques. In addition, the nonlinearities in the upcoming NRC revision can likely be addressed using the techniques described in this paper.

ACKNOWLEDGMENTS

Research was partially supported by the University of California, Davis, Sesnon Endowed Chair program, the USDA National Institute of Food and Agriculture Multistate Research Project NC-2040 (University of California–Davis), the National Institute of Food and Agriculture (Washington, DC) Project no. 2020-68014-31466, and the Innovation Center for US Dairy (Rosemont, IL). The authors have not stated any conflicts of interest.

REFERENCES

Alqaisi, O., and E. Schleck. 2021. Feeding models to optimize dairy feed rations in view of feed availability, feed prices and milk production scenarios. Sustainability (Basel) 13:215. https://doi.org/10.3390/su13010215.

Boggs, P. T., and J. W. Tolle. 1995. Sequential quadratic programming. Acta Numer. 4:1–51. https://doi.org/10.1017/S09624929000002518.

Boston, R., D. Fox, C. Sniffen, E. Janczewski, R. Munson, and W. Chalupa. 2000. The conversion of a scientific model describing dairy cow nutrition and production to an industry tool: The CPM-Dairy project. Pages 361–377 in Modeling Nutrition of Farm Animals. J. P. McNamara, J. France, and D. E. Beever, ed. CAB International.

Bozic, M., J. Newton, C. S. Thraen, and B. W. Gould. 2012. Mean-reversion in income over feed cost margins: Evidence and implications for managing margin risk by US dairy producers. J. Dairy Sci. 95:7417–7428. https://doi.org/10.3168/jds.2012-5818.

Burer, S. 2009. On the copositive representation of binary and continuous nonconvex quadratic programs. Math. Program. 120:479–495. https://doi.org/10.1007/s10107-008-0223-z.

Chandler, P. T., and H. W. Walker. 1972. Generation of nutrient specifications for dairy cattle for computerized least cost ration formulation. J. Dairy Sci. 55:1741–1749. https://doi.org/10.3168/jds.s0022-0302(72)85752-7.

Fox, D. G., L. O. Tedesc, T. P. Tylutki, J. B. Russell, M. E. Van Amburgh, L. E. Chase, A. N. Pell, and T. R. Overton. 2004. The Cornell Net Carbohydrate and Protein System model for evaluating feed nutrition and nutrient excretion. Anim. Feed Sci. Technol. 112:29–78. https://doi.org/10.1016/j.anifeedsc.2003.10.006.

France, J., and E. Keubreab. 2008. Introduction. Pages 10–11 in Mathematical Modelling in Animal Nutrition. CAB International.

Gurobi Optimization LLC. 2021. Gurobi Optimizer Reference Manual. Accessed Jul. 29, 2021. http://www.gurobi.com.

Keubreab, E., K. F. Reed, V. E. Cabrera, P. A. Vadas, G. Thoma, and J. M. Tricario. 2019. A new modeling environment for integrated dairy system management. Anim. Front. 9:25–32. https://doi.org/10.1093/af/vfz004.

Kronqvist, J., D. E. Bernal, A. Lundell, and I. E. Grossmann. 2019. A review and comparison of solvers for convex MINLP. Optim. Eng. 20:397–455. https://doi.org/10.1007/s10282-018-9411-8.

Laar, A. H., and A. G. Doig. 1960. An automatic method of solving discrete programming problems. Ecometrika 28:497–520. https://doi.org/10.2307/1910129.

Li, J. 2021. liangent/NLP-ration: The application of nonlinear programming on designing feed formulation for dairy cattle (V1.0). Zenodo. https://doi.org/10.5281/zenodo.5332918.

Morges, L. E., J. E. Wilen, P. H. Robinson, and J. G. Fadel. 2012. A linear programming model to optimize diets in environmental policy scenarios. J. Dairy Sci. 95:1267–1282. https://doi.org/10.3168/jds.2011-4651.

NRC (National Research Council). 1989. Nutrient Requirements of Dairy Cattle. 6th ed. National Academy Press.

NRC (National Research Council). 2000. Nutrient Requirements of Beef Cattle. Update 2000. National Academy Press.

NRC (National Research Council). 2001. Nutrient Requirements of Dairy Cattle. 7th ed. National Academy Press.

O’Connor, J. D., C. J. Sniffen, D. G. Fox, and R. A. Milligan. 1989. Least cost dairy cattle ration formulation model based on the degradable protein system. J. Dairy Sci. 72:2733–2745. https://doi.org/10.3168/jds.S0022-0302(89)79417-0.

Pennsylvania State University Cooperative Extension. 2020. Penn State Feed Price List. Accessed Jan. 15, 2021. https://extension.psu.edu/files/food-price-lists.

Qu, J., T. C. Hsiao, E. J. DePeters, D. Zaccaria, R. L. Snyder, and J. G. Fadel. 2019. A goal programming approach for balancing diet costs and feed water use under different environmental conditions. J. Dairy Sci. 102:11504–11522. https://doi.org/10.3168/jds.2019-16543.

Rotz, C. A., M. S. Corson, D. S. Chianese, F. Montes, S. D. Hafner, and C. U. Coiner. 2016. The Integrated Farm Systems Model: Reference Manual. USDA Agricultural Research Service.

Rotz, C. A., D. R. Mertens, D. R. Buckmaster, M. S. Allen, and J. H. Harrison. 1999. Our Industry Today: A dairy herd model for use in whole farm simulations. J. Dairy Sci. 82:2826–2840. https://doi.org/10.3168/jds.S0022-0302(99)79417-0.

Tawarmalani, M., and N. V. Sahinidis. 2013. Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, Vol. 65. Springer Science & Business Media.

Taylor, B. 2009. Introduction to Management Science. Module C, Integer Programming: The Branch and Bound Method. Pearson.

Tozer, P. R. 2000. Least-cost ration formulations for Holstein dairy heifers by using linear and stochastic programming. J. Dairy Sci. 83:443–451. https://doi.org/10.3168/jds.S0022-0302(00)74901-0.

Van Rossum, G., and F. L. Drake. 2009. Python 3 Reference Manual. CreateSpace.

Veltman, K., C. A. Rotz, L. E. Chase, J. Cooper, P. Ingram, R. C. Izaurralde, C. D. Jones, R. Gaillard, R. A. Larson, M. Ruark, W.
Salas, G. Thoma, and O. Jolliet. 2018. A quantitative assessment of beneficial management practices to reduce carbon and reactive nitrogen footprints and phosphorus losses on dairy farms in the US Great Lakes region. Agric. Syst. 166:10–25. https://doi.org/10.1016/j.agsy.2018.07.005.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovsk, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. May- orov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G.-L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Profass, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulik, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nuñez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kiømmerer, M. Bolingbrooke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, and Y. Vázquez-Baesa. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nat. Methods 17:261–272. https://doi.org/10.1038/s41592-019-0686-2.

ORCIDS

Li et al.: NONLINEAR PROGRAMMING FOR RATION FORMULATION

J. Li https://orcid.org/0000-0002-4854-2613
E. Kebreaab https://orcid.org/0000-0002-0833-1352
Fengqi You https://orcid.org/0000-0001-9609-4299
J. G. Fadel https://orcid.org/0000-0002-3512-5036
T. L. Hansen https://orcid.org/0000-0001-8778-3893
C. VanKerkhove https://orcid.org/0000-0002-2278-0461
K. F. Reed https://orcid.org/0000-0003-3936-6723