Beyond the Edge of Feasibility: Analysis of Bottlenecks

Mohammad Reza Bonyadi, Zbigniew Michalewicz, and Markus Wagner

Optimisation and Logistics
The University of Adelaide, Australia
http://cs.adelaide.edu.au/~optlog/

Abstract. The productivity of real-world systems is often limited by so-called bottlenecks. Hence, usually companies are not only interested in finding the best ways to schedule their current resources so that their benefits are maximized (optimization), but, in order to increase the productivity, they also conduct some analysis to find bottlenecks in their system and eliminate them in the most efficient way (e.g., with the lowest investment). We show that the current frequently used analysis (based on average shadow price) for identifying bottlenecks has some limitations: (1) it is limited to linear constraints, (2) it does not consider all potential sources for bottlenecks in a system, and (3) it does not provide adequate tools for decision makers to find the best way of investment to eliminate bottlenecks and maximize the profit they can gain. We propose a more comprehensive definition of bottlenecks that covers these limitations. Based on this new definition, we propose a multi-objective model for the benefit and investment. The solution for this model provides the best way of investment in resources to achieve maximum profit. As the proposed model is multi-objective and non-linear, it opens an important opportunity for the application of evolutionary algorithms, which can subsequently have a significant impact on the decision making process of companies.

Keywords: Constraints, bottlenecks, what-if analysis, feasibility.

1 Introduction

Usually real-world optimization problems contain constraints in their formulation. The definition of constraints in management sciences is "anything that limits a system from achieving higher performance versus its goal" [5]. In general, a constrained optimization problem (COP) is defined as:

$$
\begin{aligned}
&\text{find } x \in S \text{ s.t. } z = max\{f(x)\} \ \text{ subject to} \\
&\qquad g_i(x) \leq 0, \ \text{for } i = 1 \text{ to } q \\
&\qquad h_i(x) = 0, \ \text{for } i = q+1 \text{ to } m
\end{aligned}
\tag{1}
$$

where f, g_i, and h_i are real-valued functions on the search space S, q is the number of inequalities, $m - q$ is the number of equalities, and s.t. is the short form of "such that" [2, 12]. Hereafter, the term COP refers to this formulation.

G. Dick et al. (Eds.): SEAL 2014, LNCS 8886, pp. 431–442, 2014.

It is believed that the optimal solution of most real-world optimization problems is found on the edge of a feasible area of the search space of the problem [15]. This belief is not limited to computer science, but it is also found in operational research (linear programming, LP) [3] and management sciences (theory of constraints) [11, 14] articles. The reason behind this belief is that, in real-world optimization problems, constraints usually represent limitations of availability of resources. As it is usually beneficial to utilize the resources as much as possible to achieve a high-quality solution (in terms of the objective value, f), it is expected that the optimal solution is a point where a subset of these resources is used as much as possible, i.e., $g_i(x^*) = 0$ for some i and a particular high-quality x^* in the general formulation of COPs [1]. Thus, the best feasible point is usually located where the value of these constraints achieve their maximum values (0 in the general formulation). The constraints whose values are maximized at the optimum point are called *active* constraints. The constraints that are active at the optimum solution can be thought of as *bottlenecks* that constrain the achievement of a better objective value [11, 13].

Decision makers in industries usually use some tools, known as decision support systems (DSS) [8], as a guidance for their decisions in different areas of their systems. Probably the most important areas that decision makers need guidance from DSS are: (1) optimizing schedules of resources to gain more benefit (accomplished by an optimizer in DSS), (2) identifying bottlenecks (accomplished by analyzing constraints in DSS), and (3) determining the best ways for future investments to improve their profits (accomplished by an analysis for removing bottlenecks[1], known as what-if analysis in DSS). Such support tools are more readily available than one might initially think: for example, the widespread desktop application Microsoft Excel provides these via an add-in.[2]

Identification of bottlenecks and the best way of investment is at least as valuable as the optimization in many real-world problems from an industrial point of view because: “An hour lost at a bottleneck is an hour lost for the entire system. An hour saved at a non-bottleneck is a mirage” [6]. Industries are not only after finding the best schedules of their resources (optimizing the objective function), but they are also after understanding the tradeoffs between various possible investments and potential benefits. During the past 30 years, evolutionary computation methodologies (e.g., evolutionary algorithms) have provided appropriate tools for optimization. However, the last two areas (identifying bottlenecks and removing them) that are needed in DSSs seem to have remained untouched by evolutionary computation methodologies while it has been an active research area in management and operations research.

In this article, we review some existing studies on identifying and removing bottlenecks. We investigate the most frequently used bottlenecks removing analysis (the so-called average shadow price [3]) and identify its limitations.

[1] The term *removing a bottleneck* refers to the investment in the resources related to that bottleneck to prevent those resources from constraining the problem solver to achieve better objective values.

[2] `http://tinyurl.com/msexceldss`, last accessed 29th March 2014.

We argue that the root of these limitations can be found in the interpretation of constraints and the definition of bottlenecks. In particular, the previous studies have assumed only linear constraints and they have related bottlenecks only to one specific property of resources (usually the availability of resources). Further, they have not provided appropriate tools to guide decision makers in finding the best ways of investments in their system so that their profits are maximized by removing the bottlenecks. We propose a more comprehensive definition for bottlenecks that not only leads us to design a more comprehensive model for determining the best investment in the system, but also addresses all mentioned limitations. Because the new model is multi-objective and may lead to the formulation of non-linear objective functions/constraints, evolutionary algorithms have a good potential to be successful on this proposed model. In fact, by applying multi-objective evolutionary algorithms to the proposed model, the found solutions represent points that optimize the objective function and the way of investment with different budgets at the same time.

This article is structured as follows. We explain the relevant concepts in Section 2. In Section 3, we highlight limitations of a well-known bottleneck definition. In Section 4, we present our model of bottlenecks that addresses these limitations. In Section 5 we present two first evolutionary approaches that consider investments in order to remove bottlenecks. We conclude the paper in Section 6, where we also provide directions for future research.

2 Background

In this section, we provide background information on linear programming, the concept of shadow price, and bottlenecks in general. Let us begin with linear programming. A Linear Programming (LP) problem is a special case of COP (as defined in eq. 1), where $f(x)$ and $g_i(x)$ are linear functions:

$$\text{find } x \text{ such that } z = max\{c^T x\} \text{ subject to } Ax \leq b \tag{2}$$

where A is a $m \times d$ dimensional matrix known as *coefficients matrix*, m is the number of constraints, d is the number of dimensions, c is a d-dimensional vector, b is a m-dimensional vector known as Right Hand Side (RHS), $x \in \mathbb{R}^d$, and $x \geq 0$. The shadow price (SP) for the i^{th} constraint of this problem is the value of z when b_i is increased by one unit. This in fact refers to the best achievable solution if the RHS of the i^{th} constraint was larger, i.e., that more resources were available [10].

The concept of SP in Integer Linear Programming (ILP) is different from the one in LP [13]. The definition for ILP is similar to the definition of LP, except that $x \in \mathbb{Z}^d$. In ILP, the concept of Average Shadow Price (ASP) was introduced in [9]. Let us define the *perturbation function* $z_i(w)$ as follows:

$$\text{find } x \text{ s.t. } z_i(w) = max\{c^T x\} \text{ subject to } a_i x \leq b_i + w,\ a_k x \leq b_k, \forall k \neq i \tag{3}$$

where a_i is the i^{th} row of the matrix A and $x \geq 0$. Then, the ASP for the i^{th} constraint is defined by $ASP_i = sup_{w>0}\{(z_i(w) - z_i(0))/w\}$. ASP_i represents that if adding one unit of the resource i costs p and $p < ASP_i$, then it is beneficial (the total profit is increased) to buy w units of this resource. This information is very valuable for the decision maker as it is helpful for removing bottlenecks. Although the value of ASP_i refers to "buying" new resources, it is possible to similarly define a selling shadow price [9]. The concept of ASP was extended in a way that a set of resources were considered [4] rather than only one resource at a time. Note, however, that this set is predefined by the user and then the analysis is conducted [4]. There, it was also shown that ASP can be used in mixed integer LP (MILP) problems.

Now, let us take a step back from the definition of ASP in the context of ILP, and let us see how it fits into a bigger picture of resources and bottlenecks. As we mentioned earlier, constraints usually model availability of resources and limit the optimizers to achieve the best possible solution which maximizes (minimizes) the objective function [10, 11, 14]. Although finding the best solution with the current resources is valuable for decision makers, it is also valuable to explore opportunities to improve solutions by adding more resources (e.g., purchasing new equipment) [9]. In fact, industries are after the most efficient way of investment (removing the bottlenecks) so that their profit is improved the most.

Let us assume that the decision maker has the option of providing some additional resource of type i at a price p. It is clearly valuable if the problem solver can determine if adding a unit of this resource can be beneficial in terms of improving the best achievable objective value. It is, however, not necessarily the case that adding a new resource of the type i improves the best achievable objective value. As an example, consider there are some trucks that load products into some trains for transportation. It might be the case that adding a new train does not provide any opportunity for gaining extra benefit because the current number of trucks is too low and they can not fill the trains in time. In this case, we can say that the number of trucks is a bottleneck. Although it is easy to define bottleneck intuitively, it is not trivial to define this term in general.

There are a few different definitions for bottlenecks [13]. A definition for bottlenecks was proposed in [13] which was claimed to be the most comprehensive definition: "a set of constraints with positive average shadow price". In fact, the average shadow price in a linear and integer linear program can be considered as a measure for bottlenecks in a system [11].

3 Limitations of the Existing Bottleneck Definition

Although ASP can be useful in determining the bottlenecks in a system, it has some limitations when it comes to removing bottlenecks. In this section, we discuss some limitations of removing bottlenecks based on ASP.

Obviously, the concept of ASP has been only defined for LP and MILP, but not for problems with non-linear objective functions and constraints. Thus, using the concept of ASP prevents us from analyzing bottlenecks in a non-linear system.

Let us consider the following simple problem[3] (the problem is extremely simple and it has been only given as an example to clarify limitations of the previous definitions): in a mine operation, there are 19 trucks and two trains. Trucks are used to fill trains with some products and trains are used to transport products to a destination. The rate of the operation for each truck is 100 tonnes per hour (tph) and the capacity of each train is 2,000 tonnes. What is the maximum tonnage that can be loaded to the trains in one hour? The ILP model for this problem is given in eq. 4:

$$\text{find } x \text{ and } y \text{ s.t. } z = max\{2000y\} \text{ subject to} \tag{4}$$
$$g_1 : 2000y - 100x \leq 0, \quad g_2 : x \leq 19, \quad g_3 : y \leq 2$$

where $x \geq 0$ is the number of trucks and $y \geq 0$ is the number of loaded trains (y can be a floating point value which refers to partially loaded trains). The constraint g_1 limits the amount of products loaded by the trucks into the trains (trucks can not overload the trains). The solution is obviously $y = 0.95$ and $x = 19$ with objective value 1,900. ASPs for the constraints are as follows:

– ASP for g_1 is 1: by adding one unit to the first constraint ($2000y - 100x \leq 0$ becomes $2000y - 100x \leq 1$) the objective value increases by 1,
– ASP for g_2 is 100: by adding 1 unit to the second constraint ($x \leq 19$ becomes $x \leq 20$) the objective value increases by 100,
– ASP for g_3 is 0: by adding 1 unit to the third constraint ($y \leq 2$ becomes $y \leq 3$) the objective value does not increase.

Accordingly, the first and second constraints are bottlenecks as their corresponding ASPs are positive. Thus, it would be beneficial if investments are concentrated on adding one unit to the first or second constraint to improve the objective value. Adding one unit to the first constraint is meaningless from the practical point of view. In fact, adding one unit to RHS of the constraint g_1 means that the amount of products that is loaded into the trains can exceed the trains' capacities by one ton, which is not justifiable. In the above example, there is another option for the decision maker to achieve a better solution: if it is possible to improve the operation rate of the trucks to 101 tph, the best achievable solution is improved to 1,919 tonnes. Thus, it is clear that the bottleneck might be a specification of a resource (the operation rate of trucks in our example) that is expressed by a value in the coefficients matrix and not necessarily RHS. The commonly used ASP, which only gives information about the impact of changing RHS in a constraint, cannot formulate such bottlenecks.

Figure 1 illustrates this limitation. The value of ASP represents only the effects of changing the value of RHS of the constraints (Figure 1, left) on the objective value while it does not give any information about the effects the values in the coefficients matrix might have on the objective value (constraint g_1 in Figure 1,

[3] We have made several such industry-inspired stories and benchmarks available: `http://cs.adelaide.edu.au/~optlog/research/bottleneck-stories.htm`

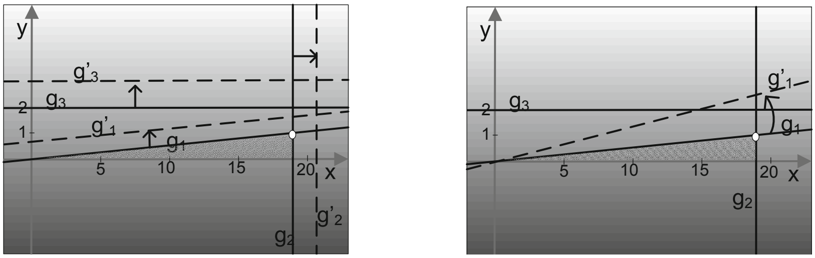

Fig. 1. x and y are number of trucks and number of trains respectively, gray gradient: indication of objective value (the lighter the better), shaded area: feasible area, g_1, g_2, g_3 are constraints, the white point is the best feasible point

right). However, as we showed in our example, it is possible to change the values in the coefficient matrix in order to achieve better solutions.

The value of ASP does not provide any information about the best strategy of selecting bottlenecks to remove. In fact, it only provides information about the benefit of elevating the RHS in each constraint or a given set of constraints and does not say anything about the order of significance of the bottlenecks. It remains the task of the decision maker to compare different scenarios by selecting different subset of constraints (also known as *what-if* analysis[4]). Of course one can analyze all possible subsets of constraints to find which subset is the most beneficial one to invest on. However, this strategy potentially leads to solving another hard problem, that is a subset selection. From a managerial point of view, it is important to answer the following question: is adding one unit to the first constraint (if possible) better than adding one unit to the second constraint (purchase a new truck)? Note that in real-world problems, there might be many resources and constraints, and a manual analysis of different scenarios might be prohibitively time consuming. Thus, a smart strategy is needed to find the best set of to-be-removed bottlenecks in order to gain maximum profit with lowest investment. In summary, the limitations of identifying bottlenecks using ASP are:

Limitation 1. ASP is only applicable if objective and constraints are linear.
Limitation 2. ASP does not evaluate changes in the coefficients matrix (the matrix A) and it is only limited to RHS.
Limitation 3. ASP does not provide information about the strategy for investment in resources, and the decision maker has to manually conduct analyses to find the best investment strategy.

[4] In the operational research community, there are related terms such as sensitive analysis and post-optimality [7].

4 A New Model for Bottleneck

In this section a new definition for bottleneck and a new model for removing bottlenecks (investment) is proposed that addresses limitations listed in Section 3.

Each constraint g_i in a real-world optimization problem usually models not only the availability of resources, but also other *specifications* of resources such as rates and capacities. Each of these specifications is encoded in a coefficient in the constraints. Accordingly, we propose a new definition for bottleneck:

Definition 1. *A bottleneck is a modifiable specification of resources that by changing its value, the best achievable performance of the system is improved.*

Note that this definition is a generalization of the definition of bottleneck in [13]: "a set of constraints with positive average shadow price is defined as a bottleneck". In fact, the definition in [13] concentrated on RHS only (it is just about the average shadow price) and it considers a bottleneck as a set of constraints, while our definition is based on any modifiable coefficient in the constraints (from capacity, to rates, or availability) and it introduces each specification of resources as a potential bottleneck. As an example, based on our definition, the operational rate of trucks can be a bottleneck, while according to the definition in [13], this is not possible [5] (see Limitation 2 in Section 3). Of course the set of all modifiable specifications need to be provided by the user.

According to the proposed bottlenecks definition, in order to invest on a part of a system to achieve maximum improvement of the objective of that system, not only RHS of all constraints should be assessed, but also all modifiable specifications in constraints need to be processed (e.g., tuning up trucks rather than buying new trucks) for potential changes. Hence, it is clear that the earlier methodologies based on ASP were not able to process all opportunities for removing bottlenecks and investments to maximize improvement in the objective function (see Limitation 3 in Section 3).

We propose a new model to address the limitations of ASP for removing bottlenecks and finding the best way of investment. We define the vector l_i which contains all modifiable specifications in the constraint g_i. For any COP in the form of eq. 1, we define a Bottleneck COP (BCOP) as follows:

$$\text{find } x \text{ and } l \text{ s.t. } z = \begin{cases} max(f(x,l)) \\ min(B(l)) \end{cases} \quad \text{subject to } g_i(x, l_i) \leq 0 \text{ for all } i \tag{5}$$

where l is a vector (l might contain continuous or discrete values) which contains l_i for all i and $B(l)$ is a function that calculates the cost of modified specifications of resources coded in the vector l. Note that in the linear case, $l_i = \{a_i, b_i\}$ where a_i is the i^{th} row of the matrix A in eq. 2. If we consider $l_i = \{b_i\}$ and

[5] One can argue that the operational rate is another constraint that can be modeled by a new variable ($g_4 : v = 100$ in eq. 4). However, if this constraint is added to the definition of the problem then constraint g_1 becomes non-linear ($g_1 : 2000y - vx \leq 0$) which then is not suitable for ASP.

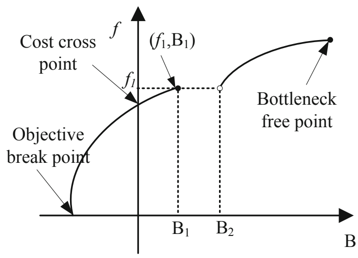

Fig. 2. The impact of an investment (B) on the achievable objective value f (maximization is the goal): positive/negative investments (reduction of resources) usually results in an increase/decrease in the best achievable objective value

$g_i(x) = a_i x - b_i$ (linear constraints) then eq.5 can express ASP_i. Figure 2 shows that, if all pieces of equipment are sold, the best achievable objective value is zero ($f = 0$) because nothing can be produced anymore (this is called "objective break" point in the figure), that is the same as the selling shadow price [9]. The "cost cross" point shows the point where the best objective value is achieved with the current specification of resources ($B = 0$). The point "bottleneck free" is the point where the optimum solution of the search space is inside the feasible region. From a practical point of view, in this situation, no matter how the decision maker invests, the profit is not improved any more. Note also that sometimes the amount of investment up to some value might not change the best achievable objective value. As an example, any investment from B_1 to B_2 does not result in any improvement in the objective value.

Let us assume that the associated solution to the point (f_1, B_1) is $\boldsymbol{x'}$ and $\boldsymbol{l'}$. This solution can be interpreted as if the decision maker invests B_1, the best way to spend this budget is to change the specifications values to $\boldsymbol{l'}$ (which costs B_1) and the best achievable objective value in this case is f_1. Note that:

- a BCOP can be formulated for linear and non-linear systems,
- any modifiable specification of resources can be formulated in a BCOP in the vector l and the values for this vector are examined by the solver,
- solutions for a BCOP contain best investment strategies for various budgets.

Any multi-objective optimization algorithm can be applied to solve a BCOP. Also, as the specifications can be coefficients in the constraints, the constraints become non-linear, which makes the problem non-linear so that linear programming methodologies are not useful in solving this problem.

Let use assume that, in the example from Section 3, the decision maker can budget \$500,000 to improve the maximum loaded products per hours into the trains. Also, the specifications that can be altered in the system are:

- the operation rate of the trucks can be increased up to 120 (i.e., $l_1 = \{100, 101, ..., 120\}$) for \$100 per tph per truck,
- the capacity of trains can be increased to 2100, with the step size 20 (i.e., $l_2 = \{2000, 2020, 2040, 2060, ..., 2100\}$ tonnes) for \$200 per ton per train,

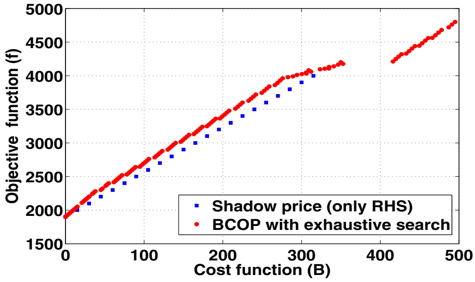

Fig. 3. Impact of an investment on the achievable objective value

– the number of trucks can be increased up to 40 (i.e. $l_3 = \{19, 20, ..., 40\}$), each truck costs \$15,000,
– the number of trains can be increased up to 5 (i.e $l_4 = \{2, 3, 4, 5\}$), each train costs \$100,000.

The BCOP for this example is written as:

$$\text{find } x \text{ and } l \text{ such that } z = \begin{cases} max\{l_2 y\} \\ min\{0.1 l_3(l_1 - 100) + 0.2 l_4(l_2 - 2000) + \\ 15(l_3 - 19) + 100(l_4 - 2)\} \end{cases} \tag{6}$$
$$\text{subject to} \qquad l_2 y - l_1 x \leq 0,\; x \leq l_3,\; y \leq l_4$$

Note that $B(l) = 0$ for $l = \{100, 2000, 19, 2\}$ (the current specification of the resources). We can solve this problem by using two methods: changing the value of RHS according to the ASP in eq. 4, or performing an exhaustive search algorithm that solves BCOP in eq. 6. Note that only practically possible were added to the list of solutions. Figure 3 shows the results.

It is clear that when BCOP is used more opportunities for investment are examined, which can potentially result in higher benefits with smaller investments. As an example, with \$135,000 investment, a solution with $f = 2950$ is found by solving BCOP with $l = \{118, 2000, 25, 2\}$. However, the best solution found based on the shadow price for \$135,000 investment was only $f = 2800$ with $l = \{100, 2000, 28, 2\}$. According to the solution of BCOP, if the decision maker is going to invest \$135,000, the best way of investment (leading to maximum improvement in the objective function) is to buy 6 new trucks ($\$6 \times 15000$) and upgrade all trucks to carry 118 tph (18 tph tune up, which costs $\$18 \times 100 \times 25$), which all together costs \$135,000. However, by using the shadow price calculations, the decision maker needs to buy 9 new trucks ($\$9 \times 15000$), which improves the objective value to 2800. It is clear that better objective values can be achieved by investing the same amount if we use BCOP.

5 An Evolutionary Algorithm for BCOP

In this section we propose two methods to solve BCOPs based on a multi-objective genetic algorithm. As the first algorithm, we use a basic algorithm

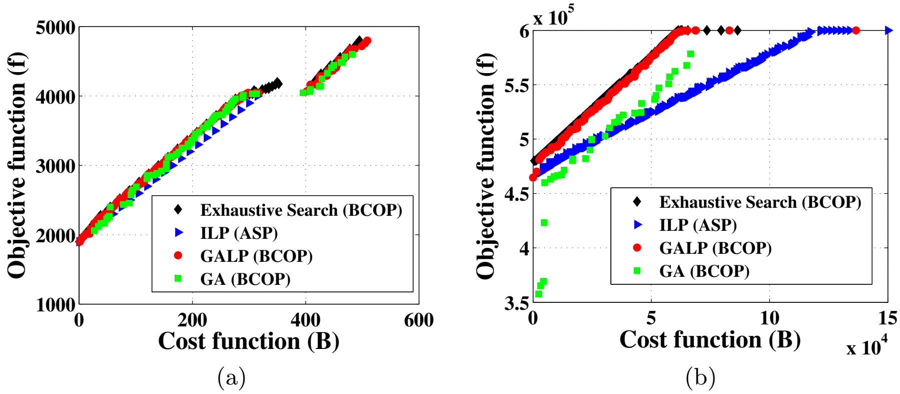

Fig. 4. Impact of an investment on the achievable objective value (a) train loading and (b) tomato farm

with tournament selection, one point crossover ($p_c = 0.9$, set via some trials), and uniform random mutation ($p_m = 0.3$, set via some trials). Each individual contains the vector l and all decision variables x. We call this first approach GA.

To handle multiple objectives, we use the following simple approach. Two solutions x_0 and x_1 are compared based on $G(x) = \sum_{i=1}^{m} max(g(x), 0)$, which is known as constraint violation value. If $G(x_0) = G(x_1)$ or $G(x_0) \leq 0$ and $G(x_1) \leq 0$, then we use the dominance relation in multi-dimensional spaces (if both are equal or non-dominating select one randomly, otherwise select the dominating one). Otherwise, we select the solution that is better in terms of constraint violation value (preferring smaller constraint violation values).

The second multi-objective algorithm GALP is based on GA. Here, the individuals contain only the vector l, and linear programming is used to find the best vector x for each generated l.

We applied both methods to the problem defined in eq. 6 with 100 individuals for 100 iterations (all non-dominating solutions found are reported in Fig. 4(a)). It is clear that both evolutionary methods have found good approximations of the Pareto front (computed by exhaustive search). It is also clear that (1) the performance of the GALP is slightly better than that of GA, and (2) our basic approaches outperform the established ASP based approach.

This means that both our approaches can be used to better plan the best investment for industries. In the following, let us consider a second example, this time from agriculture, to illustrate that our formulation is straight-forward and that it can support the decision making processes in the real-world.

Second Scenario: Agricultural Allocation.[3] A farmer owns 1,000 acres of more or less homogeneous farmland. His options are to breed cattle, or to grow wheat, corn, or tomatoes. Annually, 12,000 hours of labor are available. For simplicity, we will assume here that these can be used at any time during the year, i.e., through hiring casual labor during seasons of high need, e.g., for harvesting.

In the following, we list information regarding the profit, yield, and labor needs for the four economic activities:

– cattle: \$1,600/head profit, 0.25 heads/acre yield, 40 h/head annual labor
– wheat: \$5/bushel profit, 50 bushels/acre yield, 10 h/acre annual labor
– corn: \$6/bushel profit, 80 bushels/acre yield, 12 h/acre annual labor
– tomatoes: 50 cent/lb profit, 1,000 lbs/acre yield, 25 h/acre annual labor

It is required that at least 20% of the farmland that is cultivated in the process is used for the purpose of cattle breeding, at most 30% of the available farmland can be used for growing tomatoes, and the ratio between the amount of farmland assigned to growing wheat and that left uncultivated should not exceed 2 to 1.

Now (and this is the challenging bit), the farmer can make certain investments that can possibly increase the overall profit per year: (1) additional acres of farmland can be rented at \$200 per year, (2) additional labor can be hired at \$20 per hour, (3) a tomato packing machine can be rented for \$5,000 per year, which reduces 25 h/acre to 20 h/acres, (4) a "tomato grower's licence" can be bought for \$10,000 per year, which increases the max ratio from 30% to 35%, and (5) the value 0.25 heads/acre can be improved up to 0.7 heads/acres for \$10,000 (0.25 needs \$0, 0.7 needs \$10,000, and anything in between is linear).

The question now is: should the farmer invest, and if so, how? In Figure 4(b) we show the results of the different approaches.[6] Just as in the previous train loading example, our evolutionary approach GALP that makes use of the BCOP formulation clearly outperform the approach based on average shadow price. The results of GALP are close to those of found by an exhaustive search. Note that the approach based on average shadow price is not able to assess all cases for investment, which makes GALP more appropriate to find best investment plan. Note that in this example the methods were run for 1000 iterations.

6 Conclusions and Directions

In this paper we proposed a new definition for bottlenecks and a new model to guide decision makers to make the most profitable investment. We did this in order to narrow the gap between what is being considered in academia and industry. Our definition for bottlenecks and model for investment overcomes several of the drawbacks of the model that is based on average shadow prices:

1. It can work with non-linear constraints and objectives.
2. It offers changes to the coefficient matrix.
3. It can provide a guide towards optimal investments.

This more general model can form the basis for more comprehensive analytical tools as well as improved optimization algorithms. In particular for the latter application, we conjecture that nature-inspired approaches are adequate, due to the multi-objective formulation of the problem and its non-linearity.

[6] The construction of the BCOP formulation is straight-forward, and we omit it due to space constraints. It is available on the above-mentioned website of our *stories*.

Bottlenecks are ubiquitous and companies make significant efforts to eliminate them to the best extent possible. To the best of our knowledge, however, there seems to be very little published research on approaches to identify bottlenecks—research on optimal investment strategies in the presence of bottlenecks seems to be even non-existent. In the future, we will push this research further, in order to improve decision support systems. We will design nature-inspired single-objective and multi-objective algorithms with the goal to support decision makers to make the best possible investments in their constrained systems.

References

[1] M. R. Bonyadi and Z. Michalewicz. On the edge of feasibility: a case study of the particle swarm optimizer. In *CEC*, pp. 3059–3066. IEEE, 2014.

[2] M. R. Bonyadi, X. Li, and Z. Michalewicz. A hybrid particle swarm with velocity mutation for constraint optimization problems. In *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1–8. ACM, 2013.

[3] A. Charnes and W. W. Cooper. *Management models and industrial applications of linear programming*, Vol. 1. John Wiley and Sons, 1961.

[4] A. Crema. Average shadow price in a mixed integer linear programming problem. *European Journal of Operational Research*, 85:625–635, 1995.

[5] E. M. Goldratt. *Theory of constraints*. Nrth. River Cro.-on-Hud., NY, 1990.

[6] E. M. Goldratt and J. Cox. *The goal: a process of ongoing improvement*. Gower, 1993.

[7] P. A. Jensen and J. F. Bard. *Operations research models and methods*. John Wiley & Sons Incorporated, 2003.

[8] P. G. W. Keen. Value analysis: Justifying decision support systems. *Management Information Systems Research Center Quarterly*, 5:1–15, 1981.

[9] S. Kim and S.-c. Cho. A shadow price in integer programming for management decision. *Europ. Journal of Operational Research*, 37:328–335, 1988.

[10] T. C. Koopmans. Concepts of optimality and their uses. *The American Economic Review*, pp. 261–274, 1977.

[11] R. Luebbe and B. Finch. Theory of constraints and linear programming: a comparison. *Int. Journal of Production Research*, 30:1471–1478, 1992.

[12] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4:1–32, 1996.

[13] S. Mukherjee and A. Chatterjee. Unified concept of bottleneck. IIMA Working Papers WP2006-05-01, Indian Institute of Management Ahmedabad, Research and Publication Department.

[14] S. Rahman. Theory of constraints: A review of the philosophy and applications. *Int. Jour. of Oper. and Prod. Manag.*, 18:336–355, 1998.

[15] M. Schoenauer and Z. Michalewicz. Evolutionary computation at the edge of feasibility. In *Parallel Problem Solving from Nature (PPSN)*, Vol. 1141 of *LNCS*, pp. 245–254. Springer, 1996.