Journal of mathematics and computer science 11 (2014), 79-85

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

A Hierarchical PSO Algorithm for Solving Linear Trilevel Programming Problems

Habibe Sadeghi, Maryam Esmaeili

Department of Mathematics, Shahid Chamran University of Ahvaz, Ahvaz, Iran.

E-mail
habibe.sadeghi@mail.scu.ac.ir

m-esmaeili@phdstu.scu.ac.ir

Article history:
Received: April 2014
Accepted: May 2014
Available online : June 2014

Abstract

Trilevel programming deals with hierarchical optimization problems that in which the top-level, middle-level and bottom-level decision-makers attempt to optimize their individual objectives, but their decisions are affected by the optimal objective values presented at other levels. In this paper, we propose a hierarchical particle swarm optimization (PSO) method for solving linear trilevel programming problems (LTLPPs). The proposed method, solves the top-level, middle-level and bottom-level problems iteratively by three variants of PSO. Finally, we give some illustrative examples to show the efficiency of the proposed algorithm.

Keywords: Bilevel programming, Trilevel Programming, Particle Swarm Optimization.

1. Introduction

Multi-level programming was first defined by Candler and Townsley [5] as a generalization of mathematical programming. Many organizational decisions are made a multilevel hierarchical structure. The linear trilevel programming (LT LP) is a special case of multi-level programming and arises in many fields, including decentralized resource planning and manufacturing [6] and road network management [9]. The bilevel linear case is addressed in detail [1],[3]. In this paper, we consider a trilevel decision-making situation in which decision-maker 1 selects an action, within a specified constraint set, then decision-maker 2 selects an action within a constraint set determined by the action of decision-maker 1 and finally decision-maker 3 select an action within a constraint set determined by the action of decision makers 1 and 2. There already have beensome method for solving LTLPPs. Bard [2] and Wen and Bialas[15] give algorithms for solvingLTLPPs using cutting planes and Karush-Kuhn-Tucker (KKT) necessary optimality conditions. Benson [4] covers multilevel programming, which includes the trilevel case. Zhang and et. [16] present a Kth-best algorithm for LTLPPs. The most proposed methods require that theobjective functions at three levels TLP be differentiable or the feasible region must be convex. On the contrary, the metaheuristic needn't differentiability of objective functions, even any gradient information or the convexity of search space. As a new metaheuristic, particle swarm optimization has proved to be a competitive algorithm for solving many optimizations problems since it was proposed by Kennedy and Eberhart in 1995 [9]. Xiangyong Li et al presented a PSO algorithm for solving bilevel programming problems [10]. In this paper, we extend the algorithm presented in [10] for trilevel case. Actually, we solve a general LTLPP by solving the top-level, middle-level and bottom-level problems iteratively by three variants of PSO. The rest of paper is organized as follows. In Section 2, we state some basic definitions and theorems for LTLPs, also we introduce the standard PSO. In Section 3, we propose a hierarchical PSO algorithm for solving a general version of trilevel programming problems. Numerical example and a brief are presented in Section 4. Section 5 deals with concluding remarks.

2. Preliminaries

In this section we introduce some definitions of linear trilevel programming and the standard PSO.

2.1. Linear Trilevel Model

A basic LTLP model can be stated as follows:

For $x \in X \subseteq \mathbb{R}^n, y \in Y \subseteq \mathbb{R}^m, z \in Z \subseteq \mathbb{R}^p, f_i: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}, i = 1,2,3$

$$\begin{aligned}
&\min_{x\in X} f_1(x,y,z) = \alpha_1 x + \beta_1 y + \mu_1 z\\
&\text{s.t}\quad A_1 x + B_1 y + C_1 z \le b_1\\
&\qquad \min_{y\in Y} f_2(x,y,z) = \alpha_2 x + \beta_2 y + \mu_2 z\\
&\qquad\quad \text{s.t}\quad A_2 x + B_2 y + C_2 z \le b_2 \quad (2.1)\\
&\qquad\qquad \min_{z\in Z} f_3(x,y,z) = \alpha_3 x + \beta_3 y + \mu_3 z\\
&\qquad\qquad\quad \text{s.t}\quad A_3 x + B_3 y + C_3 z \le b_3
\end{aligned}$$

Where $\alpha_i \in \mathbb{R}^n, \beta_i \in \mathbb{R}^m, \mu_i \in \mathbb{R}^p, b_i \epsilon \mathbb{R}^{q_i}, A_i \in \mathbb{R}^{q_i \times n}, B_i \in \mathbb{R}^{q_i \times m}, C_i \in \mathbb{R}^{q_i \times p}$.

The variables *x*, *y*, *z* are called the top-level, middle-level, and bottom-level variables, and the functions $f_1(x,y,z)$, $f_2(x,y,z)$ and $f_3(x,y,z)$ are the top-level, middle-level, and bottom-level objective functions, respectively. In this model, the decision problem consists of three optimization subproblems(represented by three objective functions) in a three-level hierarchy.Now, we state some definitions and notations.

2.2. Definitions

(1) Constraint region of the LTLP:

$S = \{(x, y, z) \in X \times Y \times Z | A_i x + B_i y + C_i z \leq b_i\}$.
(2) Feasible set for the middle and bottom levels for each fixed $x \in X$:
$S(x) = \{(y, z) \in Y \times Z | B_i y + C_i z \leq b_i - A_i x, i = 2,3\}$.
(3) Feasible set for the bottom-level for each fixed $(x, y) \in X \times Y$:
$S(x, y) = \{z \in Z | C_3 z \leq b_3 - A_2 x - B_2 y\}$.
(4) Projection of S onto the top levels decision space:
$S(X) = \{x \in X | \exists (y, z) \in Y \times Z, A_i x + B_i y + C_i z \leq b_i, i = 1,2,3\}$.
(5) Projection of S onto the top and middle levels decision space:
$S(X, Y) = \{(x, y) \in X \times Y | \exists z \in Z, A_i x + B_i y + C_i z \leq b_i, i = 1,2,3\}$
(6) Rational reaction set for the middle level for $x \in S(X)$:
$P(x) = \{(y, z) \big| (y, z) \in argmin\big[f_2(x, \hat{y}, \hat{z}) \colon (\hat{y}, \hat{z}) \in S(x), \hat{z} \in argmin[f_3(x, \hat{y}, \tilde{z}), \tilde{z} \in S(x, \hat{y})]\big]\}$.
(7) Rational reaction set of the bottom-level for $(x, y) \in S(X, Y)$:
$P(x, y) = \{z | z \in argmin[f_3(x, y, \hat{z}), \hat{z} \in S(x, y)]\}$.
(8) Inducible region (IR):
$IR = \{(x, y, z) \in S | (y, z) \in P(x)\}$.
In view of the above Definitions, determining the solution to (2.1) is equivalent to solving thefollowing problem:$min\{f_1(x, y, z) | (x, y, z) \in IR\}$.
Three assumptions presented below are required to come up with the existence theorem.
(1) S is nonempty and compact.
(2) For decisions taken by the leader, the follower has some room to respond, i.e,$P(x) \neq \emptyset, P(x, y) \neq \emptyset$
(3) $P(x)$and$P(x, y)$ are point-to-point maps with respect to$x$and (x,y)respectively.

The existence of the solution for the LTLPP, can be followed from the following theorems, for proofs see [15].

Theorem 2.1. If S is nonempty and compact, then there exists an optimal solution for theLTLP problem.
Theorem 2.2. The inducible region can be expressed equivalently as a piecewise linear equalityconstraint comprised of supporting hyper planes of S.
Corollary 2.1.A solution to the LTLP problem stated in (2.1) occurs at a vertex of the IR.
Theorem 2.3. The solution$(x^*, y^*, z^*)$of the linear trilevel programming problem occurs at avertex of S.

Corollary2.2. If(x, y, z) is an extreme point of the IR, then it is an extreme point of S.

2.3. Particle Swarm Optimization

PSO is a population-based stochastic optimization algorithm. The system is initialized witha population of random solutions and searches for optima by updating generations. However,unlike GA, the PSO algorithm has no evolutionary operators, such as crossover and mutation.In the PSO algorithm, the individuals who called particles, manipulate their trajectories towardthe best region of their own previous best performance and toward the locations found by members in their neighborhoods. The location of particle$i$is represented as$X_i = (x_{i1}, \dots, x_{iD})$which is a D-dimentional vector in problem space, and its performance is evaluated on the predefined fitness function related to the problem and each particle keeps the memory of its previous best position, P_{best}. The velocity of each particle, represented as$V_i = (v_{i1}, \dots, v_{iD})$ and theposition of the particle with the best performance in the search space is represented byP_g.Theparticle velocities in each dimension are controlled by a maximal velocity, V_{max} , and the velocityin that dimension is limited to V_{max}. The flying direction of particle is the dynamical interactionof individual and social flying experience. The position change of each particle is a function of itscurrent velocity vector, the stochastically weighted difference between its current position andthe best position found by itself so far, (P_{best}) and difference between the individual's currentposition and the best position found by any member in its neighborhood (P_g). The velocity and position of *j*-th component of *i*-th particle at iteration *t* is updating by the following two equations:

$$v_i^j(t+1) = wv_i^j(t) + c_1 rand_1\left(p_i^j(t) - x_i^j(t)\right) + c_2 rand_2\left(p_g^j(t) - x_i^j(t)\right)$$

$$x_i^j(t+1) = x_i^j(t) + v_i^j(t) \qquad (2.2)$$

Where p_i^j is the *j*-th component of the best position encountered by the *i*-th particle so far, p_g^j represents the *j*-th component of the position of the best performance in whole swarm, *t*is the iteration counter,c_1 and c_2 are the acceleration coefficients; $rand_1$ and$rand_2$ are two randomnumbers in [0,1] and,*w*is inertia weight. From the velocity update equation is clear that c_2 regulates the maximum step size in the direction of the global best particle, and c_1 regulatesthe step size in the direction of the personal best position of that particle.

2.3.1 Rate of convergence improvements

Several techniques have been proposed for improving the rate of convergence of the PSO. Someof the earliest modifications to the original PSO were aimed at further improving the rateof convergence of the algorithm. Shi and Eberhart investigated the effect of*w*values in therange [0,1.4], as well as varying*w*over time [13]. Their results indicate that choosing $w \in [0.8,1.2]$ results in faster convergence, but that larger*w*value (> 1.2) results in more failuresto converge. Further empirical experiments have been performed with an inertia weight set todecrease linearly from 0.9 to 0.4 during the course of a simulation [12]. This setting allows thePSO to explore a large area at the start of the simulation run, and to refine the search later byusing a smaller inertia weight. Some research have found out that setting$c_1=c_2=2$ gets thebest overall performance. Suganthan method shows that small cognitive coefficient and largesocial coefficient can improve the algorithm convergence [14]. For more information about theanalysis of the parameter selection in PSO, see [9].

3. Proposed Algorithm for Solving Linear trilevel Programming Problems

In this section, we introduce a hierarchical PSO algorithm for solving linear trilevel programming problems. As mentioned above, LTLPs are hierarchical and sequential optimization problems. Moreover each top-level, middle-level and bottom-level problem can be considered as an individual optimization problem. In view of the characteristics of sequential decision, we can constructa hierarchical algorithm based on three variants of standard PSO to solve LTLPPs.By Theorem 2.3, the optimal solution of LTLPP occurs in an extreme point of the setS.So, the set ofScan be considered as our search space. Since often it is difficult to constructS,we consider an ordered polyhedral which covers S, as our search space. In first step we generatesome particles and their corresponding velocities, and to apply them as input values of the toplevel PSO (PSO-T), that finds the optimal solution of$f_1$in constraint region S. Let (x_0, y_0, z_0) be the output of PSO-T, In the next step we generate some new particles with fixed $x=x_0$,anduse them as input of PSO-M which finds the optimal solution of$f_2$in the feasible set for themiddle and bottom levels for fixed $x=x_0$, i.e, S(x_0), let (x_0, y_1, z_1) be the output of PSO-M.Again produce some new particles with fixed $x=x_0$, $y = y_1$, and use them as input data of PSO-B which finds the optimal solution of

f_3 in the feasible set for the bottom level, i.e., $S(x_0,y_1)$ The output of this algorithm is considered as the solution of LTLPP.

4.Computational Experiences

In order to test the proposed PSO algorithm, we solved two problems with this method. Thecodes have been written in Matlab 7.1. Numerical experiments have been carried out on a Pentium (R) 2.00 GHz processor. For each problem, 30 runs were simulated. This was done to ensure that the randomness of generating the particles, did not have a major influence onthe final solution. The parameters for the implementation of the algorithm are set as follows: The swarm sizes N_{max} are set to 40, 30 and 20, for PSO-T, PSO-M and PSO-B respectively.The number of maximum generations, G_{max} are set to 60, 40, 20, for PSO-T, PSO-M, PSO-Brespectively, acceleration coefficient c_1 is 0.5 and, c_2 is 1.5, inertia weight, W, is set to decrease linearly from 1.2 to 0.1. For each testing problem, it can be seen that the standard deviation of the best top level objective value over 30 runs, and the difference between the best and the worst value of the top level objective, is almost equal to 0, which means that the robustness of ourproposed algorithm is very high.

We can observe relocation of particles in all iterations in PSO-T, PSO-M and PSO-B subalgorithms for testing problem one in Figures 1, 2 and 3. The summery of the found results are reported in Table 1. If we solve the problems with existent algorithms such as Kth-Best algorithm, we observe that, the solution obtained for test problem one is exact and, the solution obtained for the test problem two is very close to the exact solution in which the value of objective functions is f_1= *-35.33*, f_2= -20, f_3= *-12*.

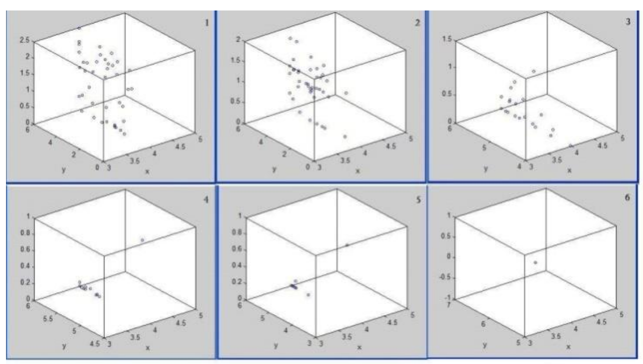

Figure 1. Relocation of particles in PSO-T

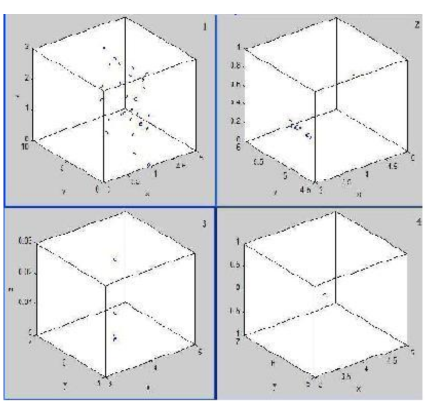

Figure 2.Relocation of particles in PSO-M

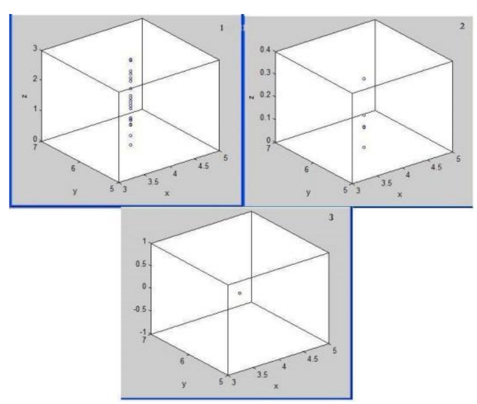

Figure 3. Relocation of particles in PSO-B

Table 1. Results for Testing Problems

	Best f_1	**Worst f_1**	**Avg.**	**Std.**	**Best f_2**	**Worst f_2**	**Best f_3**	**Worst f_3**
Test1	-20	-20	-20	0	10	10	-8	-8
Test2	-35.319	-33.5541	-34.9829	0.2124	-20	-19.0764	-12	-12

5. Conclusion

In this paper, we extend the application of PSO to solving linear trilevel programming problems. Actually, we solve a general LTLPP by solving the top-level, middle-level and bottom-levelproblems iteratively by three variants of PSO, that are called PSO-T, PSO-M and PSO-B. Withthis approach, we can solve this kind of programming problems without need to solve differentsimplex tables and without any transformation of the objective or constraints functions. Alsobecause of nature of PSO algorithm which is designed for solving nonlinear programming withoutany specified assumptions and conditions, it seems that different classes of LTLPPs can be solved more effectively through such an interaction between three variants of PSO.

6.Appendix: Testing problems

1. Testing problem one:

$$\min_{x\geq 0} f_1(x,y,z) = x - 4y + 2z$$
$$-x - y \leq -3$$
$$-3x + 2y - z \geq -10$$

$$\min_{y\geq 0} f_2\,(x,y,z) = x + y - z$$
$$-2x + y - 2z \leq -1$$
$$2x + y - 4z \leq 14$$

$$\min_{z\geq 0} f_3(x,y,z) = x - 2y - 2z$$

$$2x - y - z \leq 2$$

2. Testing problem two:

$$\min_{x\geq 0} f_1(x,y,z) = -4x + 2y - 5z$$
$$3x - y + z \leq 12$$
$$x \geq 2$$

$$\min_{y\geq 0} f_2(x,y,z) = y - 4z$$
$$2y - z \geq 2$$
$$3y + z \leq 24$$

$$\min_{z\geq 0} f_3(x,y,z) = -2z$$
$$z \leq 6$$

7. References

[1] G. Anandalingam, T. Fries, *HierarchicalOptimization: an introduction,* Annals of Operations Research, 34 (1992) 1-11.

[2] J. Bard, *An investigation of the linear three level programming problems*, IEEE Transactionson systems, Man and Cybernetics, 14(1984) 711-717.

[3] J. Bard, *Practical Bilevel Optimization, Algorithms and Applications*, Kluwer Academic Publishers, Dordrecht, London, 1998.

[4] H. Benson, *On the structure and properties of a linear multilevel programming problem,* Journal of Optimization Theory and Application, 9(1989) 353-373.

[5] W. Candler and R. J. Townsley, *A linear multilevel programming problem*, Computer and operations Research, 9(1982) 59-67

[6] D. Cao, M. Chen, *Capacitated plant selection in decentralized manufacturing environment: A bilevel optimization approach,* European journal of operational research, 169 (2006) 97-110.

[7] C. Feng, C. Wen, *Bilevel and multi objective model to control trafficflow into the disaster area ,post-earthquake*, Journal of the EasternAsia Society for Transportation Studies , 6 (2005) 4253-4268.

[8] S.R. Hejazi, *Methods for solving Linear Multilevel Programming Problems*, A Thesis Presented for the Degree of Ph.D. in Industrial Engineering, School of Engineering, Tarbiat Modarress University, Iran, 2001.

[9] J. Kennedy, R.C. Eberhart, *Particle Swarm Optimization*, IEEE International Conferenceof NeuralNetworks, Perth, Australia, (1995) 1942-1948.

[10] X. Li, P. Tian, X. Min,*AHierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems*, Lecture Notes in Computer Science , 4029 (2006) 1169-1178.

[11] R. Shakerian, S. H. Kamali, M. Hedayati, M. Alipour, *Comparative Study of Ant Colony Optimization and Particle Swarm optimization for Gris Scheduling,* TJMCS , 3(2011) 469-474

[12]Y. Shi and R.C. Eberhart, *A Modified Particle Swarm Optimizer*. IEEE International Conference of Evolutionary Computation, 1998.

[13] Y. Shi and R.C. Eberhart, *Empirical Study of Particle Swarm Optimization*. In Proceeding of the Congress on Evolutionary Computation, (1999) 1945-1949.

[14] P.N. Suganthan, *Particle Swarm Optimizer with Neighborhood Operator*, Congress on Evolutionary Computation, Washington, (1999) 1958-1962.

[15] U. Wen and W. Bialas, *The hybrid algorithm for solving the three level programming problem,* Computer and Operation Research, 13 (1986) 367-377.

[16] G. Zhang, J. Lu, J. Montero, Y. Zeng, *Model, solution concept, and K-th Best algorithm for linear trilevel programming,* Information Sciences, 180 (2010) 481-492.