Solution of Economic Lot Scheduling Problem: A Hybrid Meta-Heuristic Approach

Syed HasanAdil[1], Syed Saad Azhar Ali[2], Aarij Hussaan[1], Kamran Raza[1]

[1.]Department of Computer Science, [2.]Department of Electronic Engineering, Iqra University,
Main Campus: Defence View, Shaheed-e-Millat Road (Ext.) Karachi-75500, Pakistan
hasan.adil@iqra.edu.pk

Abstract: In this paper we suggest a hybridization scheme to solve Economic Lot Scheduling Problem (ELSP) using basic period approach. We proposed a hybrid approach based on Tabu Search (TS) optimization to find the optimum value of k_i*'s* and Golden Section Search (GSS) with parabolic interpolation to find the optimum value of basic period *T*. The proposed hybridized scheme is compared with the best known Genetic Algorithm (GA) [4] on Bomberger's dataset [1]. This hybrid approach is found competitive and efficient in solving Economic Lot Scheduling Problem and outperform the Genetic Algorithm on problems with higher machine utilization.
[Adil SH, Ali SSA, Hussaan A, Raza K. **Solution of Economic Lot Scheduling Problem: A Hybrid Meta-Heuristic Approach**. *Life Sci J* 2013; 10(3):303-309] (ISSN: 1097-8135). http://www.lifesciencesite.com. 48

Keywords: Economic Lot Scheduling Problem; Basic Period Approach; Tabu Search; Golden Section Search.

1. Introduction

The Economic Lot Scheduling Problem (ELSP) has been under research for more than four decades. The problem is computationally very complex and has been classified as NP-hard problem [1]. Despite its complexity the ELSP has been encountered in most production planning scenarios. Due to NP hard nature of the problem many researchers have developed heuristic solutions to the problem. There are four approaches to solve the ELSP problem: common cycle [7]; basic period [4, 18]; extended basic approach [3]; and time varying lot size approach [6].

As the ELSP is generally viewed as NP-hard, the focus of most research efforts has been towards generating near optimal repetitive schedule(s). To date, several heuristic solutions [4, 9, 10, 11, 12, 18] have been proposed using any one of the common cycle, basic period, extended basic approach, or time-varying lot size approaches. The common cycle approach always produces a feasible schedule and is the simplest to implement, however, in some cases the solution when compared to the lower bound is of poor quality [16]. Unlike the common cycle approach, the basic period approach allows different cycle times for different products, however, the cycle times must be an integer multiple of a basic period. Although the basic period approach generally produces a better solution to ELSP than common cycle approach, but getting a feasible schedule is NP-hard [1]. The BP approach assumes that the production runs of all products shall be made in each basic period. Then, the basic period must be long enough to accommodate the production of all the products. This is rather a restrictive condition which usually results in suboptimal solutions. The extended basic period approach removes this restriction and admits the possibility that in any basic period only a subset of the products shall be produced [14, 15, 18]. This obviates the waste of capacity of the production facility. Lastly, the time-varying lot size approach is more flexible than the other two approaches, allowing for different lot sizes for the different products in a cycle [16]. Dobson [6] showed that the time-varying lot size approach always produced a feasible schedule as well as giving a better quality solution.

The proposed research is motivated by the recent success [4, 9, 10, 11, 12, 18] of the meta-heuristics to solve ELSP. Therefore, this research investigates the use of meta-heuristics to solve the ELSP problem using basic period approach. We applied Tabu Search (TS) with Golden Section Search (GSS) [18] to find the solution and compared with existing Genetic Algorithm (GA) [4] based best known solution. The two meta-heuristics will be compared in order to calibrate their performance in regards to solution quality produced and computation time needed.

2. Basic Period Approach to ELSP

We present ELSP model [1] which is based on the basic period approach. We have to produce *m* distinct products on single production facility with the following assumptions.

- The competing products for production facility do not have any precedence over each other.
- Back-orders are not allowed.
- An item is considered for production only if its inventory is depleted to the zero level. This rule is known as Zero-Switching-Rule (ZSR).
- The production facility is assumed to be failure free and to always produce perfect quality products. The solution of the ELSP is based on

specifying an inventory cycle for each part, subject to following conditions:

- The quantity of a part produced during its cycle must be sufficient to meet demand over the cycle.
- The length of the cycle must be sufficient to permit the production of other parts scheduled during the cycle.

A schedule is feasible if the above conditions are met. This feasible solution becomes optimal if the total cost is minimized.

The following notations and equations (1-14) are used to find the solution of ELSP [1, 5]:

i	:	An item index, i={1,2, …,n}
D_i	:	Annual demand for item i (units/ year)
P_i	:	Annual production rate for item i (units/year)
H_i	:	Holding cost for item i ($/unit-year)
S_i	:	Setup cost for item i ($/setup)
τ_i	:	Setup time for item i (years)
Q_i	:	Production quantity for item i, a decision variable (units)
T_i	:	Cycle time for item i, a decision variable (in days)
TC_i	:	Total annual holding and setup cost for item i ($/year)
TC	:	Total annual holding and setup cost for all item ($/year)

The total cost for an item i is:

$$TC_i = \frac{Q_i}{2}\left(1 - {}^{D_i}/_{P_i}\right) H_i + \left({}^{D_i}/_{Q_i}\right) S_i \qquad (1)$$

The total annual cost of all n items is:

$$TC = \sum_{i=1}^{n}\left[\frac{Q_i}{2}\left(1 - {}^{D_i}/_{P_i}\right) H_i + \left({}^{D_i}/_{Q_i}\right) S_i\right] \qquad (2)$$

The ELSP is formulated as follows:

Minimize TC

$$\text{Subject to} \sum_{i=1}^{n}\left(\left({}^{D_i}/_{P_i}\right)\tau_i + \frac{D_i}{P_i}\right) \le 1 \qquad (3)$$

No two items are produced at the same time (4)

The first constraint ensures that the time spent setting up the machine and producing the items does not exceed the time available. Solving the unconstrained problem results a loose lower bound known as the independent solution (IS).

The optimal order quantity for item i is given by:

$$Q_i^* = \sqrt{\frac{2\,D_i S_i P_i}{H_i\left(1 - {}^{D_i}/_{P_i}\right)}} \qquad (5)$$

Substituting from equation (5) into equation (2) gives IS lower bound on the ELSP as follows:

$$TCIS = \sum_{i=1}^{n}\sqrt{\frac{2\,D_i S_i H_i}{(P_i - D_i)P_i}} \qquad (6)$$

Alternatively, a tighter lower bound (TCL) can be obtained by minimizing the total cost (TC) subject to constraint in equation (3):

$$Q_i^* = \sqrt{\frac{2\,D_i P_i (S_i + \lambda\,\tau_i)}{H_i (P_i - D_i)}}. \qquad (7)$$

And satisfying:

$$\lambda\left(\sum_{i=1}^{n}\frac{\tau_i D_i}{Q_i} + \sum_{i=1}^{n}\frac{D_i}{P_i} - 1\right) = 0 \qquad (8)$$

In case if the production facility in under-utilized, the capacity constraint will not be binding and TCL will be same as TCIS. However, with the higher utilization, TCL is higher than the IS lower bound. The increase in TC and TCL relative to TCIS at high utilization is due to production quantities becoming larger to reduce the time spend on setup, which substantially increases the holding cost.

Now, we discuss an analytical approach which allows achieving the optimal solution to a restricted version of the original problem mentioned in [6, 19]. The approach is called basic period approach. In basic period approach, the cycle time for every item i is an integer multiple k_i of a fundamental cycle T. Thus, the cycle time for an item i is:

$$T_i = k_i T \qquad (9)$$

Also the production quantity for an item i will becomes:

$$Q_i = T_i D \qquad (10)$$

The total cost incurred under basic period approach (TCBP) is obtained from substituting T_i and Q_i into equation (2). Thus, the total cost is:

$$TCBP = \sum_{i=1}^{n} T k_i D_i \left(1 - \frac{D_i}{P_i}\right)\frac{H_i}{2} + \frac{S_i}{T\,k_i} \qquad (11)$$

TCBP established in Equation (11) is now a function of T and k_i's. Once TCBP is established, the ELSP under BP approach is:

Minimize TCBP

$$\text{Subject to} \sum_{i=1}^{n}\left(\tau_i + \frac{D_i\, T\, k_i}{P_i}\right) \leq T \qquad (12)$$

The constraint in the above optimization problem ensures that the fundamental cycle is long enough to accommodate the production of all items even though not every item has to be produced during every fundamental cycle. The constraint guarantees the feasibility but may result in a suboptimal solution to the original problem. In [1], it is shown that the above problem can be formulated and solved as a Dynamic Programming (DP) problem. The main idea of [1] was to fix T, and solve the DP problem to obtain the optimal k_i's and then use the information to get a better estimate of the optimal T. Thus, this approach requires solving a number of DP problems to find the optimal T.

In a nutshell this approach requires a one-dimensional search on T. In each of the iteration of the search, a DP problem must be solved. Thus, a more precise estimate of the optimal T requires larger number of the DP problems to be solved that makes the use of meta-heuristics even more attractive alternate to solve the problem. The above formulation very well suits meta-heuristics. GA [4] suggested that both the T and k_i's are simultaneously determined leaving no need to solve DP problems repeatedly with different values of T. Furthermore, the curse of dimensionality due to DP is not encountered in using GA.

3. Proposed Hybridized Approach

In this research; we suggest hybridization of Tabu Search with GSS to solve ELSP using basic period approach. We have used Tabu Search to find the optimum value of integer multiple k_i's and GSS to find the optimum value of basic period T. The proposed hybridized scheme is analyzed using Bomberger's dataset [1]. A discussion on GSS is presented in [18]. GSS is incorporated with Cuckoo Search, Particle Swarm Optimization and Simulated Annealing to solve ELSP [18].

4. Tabu Search

Fred Glover proposed Tabu Search (TS) [10, 17], to allow Local Search (LS) methods to overcome local optima. It includes short term memory to prevent the reversal of recent moves, and longer term frequency memory to reinforce attractive components. The basic principle of TS is to pursue LS whenever it encounters a local optimum by allowing non-improving moves; cycling back to previously visited solutions is prevented by the use of memories, called tabu lists, that record the recent history of the search, a key idea that can be linked to Artificial Intelligence concepts. TS approach is similar to steepest ascent/mildest descent approach. Glover described TS as a Meta-Heuristic approach, i.e., a general strategy for guiding and controlling "inner" heuristics specifically tailored to the problems at hand. The basic pseudo-code of this algorithm is shown below:

Notations

S, The current solution,
*S**, The best-known solution,
*f**, Value of *S**,
N(S) The neighborhood of *S*,
Ñ(S) *The "admissible" subset of N(S) (i.e., non-tabu or allowed by aspiration).*

Initialization

Choose (construct) an initial solution *S0.*
Set $S := S_0$, $f^* := f(S_0)$, $S^* := S_0$, $T := \varnothing$.

Search

While *termination criterion not satisfied* do
  Select *S* in *argmin*[*f(S')*];
  S'∈ *Ñ(S)*
  If *f(S)* < *f**, then set *f** := *f(S)*, *S** := *S*;
    record tabu for the current move in
    T (delete oldest entry if necessary);
end while.

A. Proposed GSS-TS Hybridization Scheme

The proposed hybridized TS with GSS algorithm is discussed below:

- The nonlinear objective function given in equation (11) is minimized subject to constraint given in equation (12).
- Lower and upper bounds of T and k_i's are computed using following equations [5],

$$T^{LB} = \sum_{i=1}^{n} 0.25\, Q_i^*/P_i \qquad (13)$$

$$T^{UB} = \max\left\{ \sqrt{(2(\sum_{i=1}^{n} S_i)/\sum_{i=1}^{n} H_i D_i (1 - D_i P_i)),} \atop (\sum_{i=1}^{n} \tau_i)/(1 - D_i P_i) \right\} \qquad (14)$$

$$k_i^{LB} = 1 \qquad (15)$$

$$k_i^{UB} = \left\lceil (5\, (Q_i^*/D_i)/T)(\sum_{i=1}^{n} \frac{D_i}{P_i}) \right\rceil, \qquad (16)$$

$$i = 1, 2, \ldots, n$$

- k_i*'s* are initialized randomly between $[k_i^{LB}, k_i^{UB}]$, $i = 1, 2, \ldots, n$
- Given the initial k_i*'s*, the TCBP subject to constraint (12) can be minimized by performing a one dimensional search on T based on GSS [18].
- The following steps are repeated until the maximum iteration becomes reached or until a convergence criterion has been met.
- TS algorithm is applied to generate the new solution in k-dimensional search space. Here, each dimension represents one k_i and the whole solution represents one complete possible solution to ELSP problem.
- k_i*'s* associated with solution that do not fulfill lower and upper bound requirements are updated with randomly generated values between $[k_i^{LB}, k_i^{UB}]$.
- Newly generated k_i*'s* associated with solution in k-dimensional search space are given to apply one dimensional search on T based on GSS [18] to minimize TCBP subject to constraint (12).
- Current best k_i*'s* and T are updated that minimize the TCBP.

5. Results

The proposed hybrid TS optimization scheme is applied to solve ELSP problem using basic period approach. We use the Bomberger's dataset which is the most commonly used in the ELSP literature [1] as shown in Table 1. All the simulations are run on a year horizon and by assuming that ten items are produced on a single machine. Production activity is assumed to be 240 days in a year, only on weekdays. The results obtained from detailed analysis are exhibited in Table 2, Table 3, and Table 4.

Table 1: Data of Bomberger's problem [4, 18]

Product index, i	1	2	3	4	5	6	7	8	9	10
Base Demand	24,000	24,000	48,000	96,000	4800	4800	1440	20,400	20,400	24,000
Setup cost (S_i): \$	15	20	30	10	110	50	310	130	200	5
Production rate (P_i): units/day	30,000	8000	9500	7500	2000	6000	2400	1300	2000	15,000
Setup time (τ_i) : h	1	1	2	1	4	2	8	4	6	1
Holding cost (H_i): \$/unit-year	0.00065	0.01775	0.01275	0.01000	0.27850	0.02675	0.15000	0.59000	0.09000	0.00400

Table 2 compares the cost obtained by solving [1] problem as shown in Table 1 using TS and GA [4] algorithms. In Table 3 relative deviation from tighter lower bound (TCL), improvement achieved through TS algorithm over results obtained through GA algorithm [4], efficiency in terms of execution time taken by TS algorithm are compared while Table 4 compares the detailed solution found by TS with GA solution [4].

Table 2 shows that 48% of TS solutions are either better or similar to best results obtained from GA algorithm. On the contrary 41% of GA results are better or similar to best results obtained from TS algorithm. Therefore, it can be concluded that in majority of cases TS performed better than GA algorithm.

Table 3 shows that average relative deviation from TCL is 19.841% using TS and worst average relative deviation from TCL is 21.261% using GA algorithm, average improvement over GA is 0.952% using TS, and average CPU utilization time is 2.609 seconds using TS. It is also important to note that GA differs with TS algorithm for high utilization as well as for low utilization cases. GA found worst relative deviation from TCL for higher utilization but results for lower utilization cases are comparatively closed to TS algorithm.

Table 4 shows detailed comparison of values for T and k_i (i.e., i=1,2,…10) using TS with GA algorithm. For low utilization cases 50 to 92 k_i have different values but for high utilization cases 95 to 99 all k_i have same value '1'. TS found same value for T and k_i which gives low deviation from TCL. GA found the same value for k_i but failed to find value for T similar to TS algorithm and therefore resulting in high deviation from TCL.

Table 2: Comparison of TSIS, TCL, GA and TS solutions for Bomberger's problem [1, 4, 18]

Utilization (%)	TSIS	TCL	GA	TS	Best Cost	Best Algorithm(s)
50	5960.445	5960.445	6038.410	6036.748	6036.748	TS
55	6218.253	6218.253	6328.670	6372.022	6328.670	GA
60	6459.905	6459.905	6621.750	6619.799	6619.799	TS
65	6687.131	6687.131	6914.700	6914.837	6914.700	GA
66.18	6738.810	6738.810	7024.110	7024.100	7024.100	TS
70	6901.335	6901.335	7395.460	7395.460	7395.460	GA, TS
75	7103.674	7103.674	7789.630	7917.524	7789.630	GA
80	7295.114	7295.114	8096.010	8181.051	8096.010	GA
83	7405.090	7405.090	8250.290	8342.896	8250.290	GA
86	7511.593	7511.593	8553.310	8483.945	8483.945	TS
88.24	7588.934	7588.934	8782.420	8782.289	8782.289	TS
89	7614.763	7614.763	8874.550	8874.803	8874.550	GA
92	7714.729	7714.729	9745.800	9746.356	9745.800	GA
95	7811.608	8418.885	12018.080	11949.646	11949.646	TS
97	7874.534	11290.966	17143.000	17134.260	17134.260	TS
98	7905.510	15681.535	24533.820	24457.541	24457.541	TS
99	7936.166	29942.667	55544.470	47550.735	47550.735	TS

Table 3: Comparison of Relative Deviation from TCL, Improvement over GA, and CPU time taken by algorithms for Bomberger's problem [1, 4, 18].

	% Relative Deviation from TCL		% Improvement over GA	CPU time (sec.)
Utilization (%)	GA	TS	TS	TS
50	1.308	1.280	0.028	3.232
55	1.776	2.473	0	2.4
60	2.505	2.475	0.029	2.875
65	3.403	3.405	0	2.936
66.18	4.234	4.234	0	2.624
70	7.160	7.160	0	2.159
75	9.656	11.457	0	2.46
80	10.979	12.144	0	2.809
83	11.414	12.664	0	2.685
86	13.868	12.945	0.811	2.729
88.24	15.727	15.725	0.001	2.239
89	16.544	16.547	0	1.946
92	26.327	26.334	0	2.663
95	42.751	41.939	0.569	2.478
97	51.829	51.752	0.051	2.559
98	56.450	55.964	0.311	2.58
99	85.503	58.806	14.392	2.985
Average	21.261	19.841	0.952	2.609
Min.	1.308	1.280	0	1.946
Max.	85.503	58.806	14.392	3.232
Std. Dev.	23.939	19.765	3.471	0.320

Table 4: Detail comparison of GA and TS results for Bomberger's problem [1, 4, 18]

Utilization	Meta-heuristic	
	GA	TS
50	T = 28.183 k_i=[5,1,2,1,2,4,10,1,3,1]	T = 28.594 k_i=[5,1,2,1,2,5,9,1,3,1]
55	T = 28.762 ki=[5,2,2,1,2,4,8,1,2,1]	T = 29.314 k_i=[3,1,1,1,2,4,8,1,3,1]
60	T = 28.863 k_i=[4,1,1,1,2,4,9,1,2,2]	T = 28.798 k_i=[3,2,1,1,2,4,8,1,2,1]
65	T = 30.828 k_i=[2,1,1,1,2,3,7,1,2,1]	T = 30.838 k_i=[2,1,1,1,2,3,7,1,2,1]
66.18	T = 30.443 k_i=[2,1,1,1,2,2,6,1,2,1]	T = 30.449 k_i=[2,1,1,1,2,2,6,1,2,1]
70	T = 33.42 k_i=[2,1,1,1,1,2,3,1,2,1]	T = 33.42 k_i=[2,1,1,1,1,2,5,1,2,1]
75	T = 31.794 k_i=[3,1,1,1,2,3,7,1,1,1]	T = 35.719 k_i=[2,1,1,1,1,2,6,1,1,1]
80	T = 34.438 k_i=[2,1,1,1,1,3,6,1,1,1]	T = 35.614 k_i=[1,1,1,1,1,2,5,1,1,1]
83	T = 34.951 k_i=[1,1,1,1,1,2,5,1,1,1]	T = 35.815 k_i=[1,1,1,1,1,2,4,1,1,1]
86	T = 37.131 k_i=[1,1,1,1,1,1,5,1,1,1]	T = 38.371 k_i=[1,1,1,1,1,2,4,1,1,1]
88.24	T = 38.442 k_i=[1,1,1,1,1,1,3,1,1,1]	T = 38.436 k_i=[1,1,1,1,1,1,3,1,1,1]
89	T = 41.748 k_i=[1,1,1,1,1,1,3,1,1,1]	T = 41.758 k_i=[1,1,1,1,1,1,3,1,1,1]
92	T = 53.904 k_i=[1,1,1,1,1,1,2,1,1,1]	T = 53.914 k_i=[1,1,1,1,1,1,2,1,1,1]
95	T = 75.809 k_i=[1,1,1,1,1,1,1,1,1,1]	T = 75 k_i=[1,1,1,1,1,1,1,1,1,1]
97	T = 125.08 k_i=[1,1,1,1,1,1,1,1,1,1]	T = 125 k_i=[1,1,1,1,1,1,1,1,1,1]
98	T = 188.14 k_i=[1,1,1,1,1,1,1,1,1,1]	T = 187.5 k_i=[1,1,1,1,1,1,1,1,1,1]
99	T = 439.45 k_i=[1,1,1,1,1,1,1,1,1,1]	T= 375 k_i=[1,1,1,1,1,1,1,1,1,1]

6. Conclusion

This research presented hybridized scheme based on Tabu Search and Golden section search to solve the ELSP problem under basic period approach. This hybrid technique used Tabu Search to find the optimum value of k_i*'s* and followed by golden section search to find the basic period T. The feasibility of the solution is guaranteed with a constraint that ensures the items assigned in each period can be produced within the length of the period. The experimental results indicate following outcomes:

- The hybridization scheme was able to find BP solution comparatively similar to GA [4] for the low utilization problems.
- The hybridization scheme was also able to find comparatively better BP solutions than GA [4] for the high utilization problems.

Acknowledgements:

The authors acknowledge the support of Faculty of Engineering, Sciences & Technology, Iqra University for providing research facilities and financial assistance.

Corresponding Author:
Syed Hasan Adil
Department of Computer Science
Main Campus, Iqra University
Defence View,
Shaheed-e-Millat Road (Ext.)
Karachi-75500, Pakistan
E-mail: hasan.adil@iqra.edu.pk

References

[1] Bomberger E. A dynamic programming approach to a lot size scheduling problem. Management Science 1966; 12(11): 778–84.

[2] Elmaghraby SE. The Economic Lot Scheduling Problem (ELSP): Review and Extensions. Management Science 1978; 24(6), 587–98.
[3] Elmaghraby SE. An Extended Basic Period Approach to the Economic Lot Scheduling Problem (ELSP). Production and Industrial Systems: Future Development and the Role of Industrial and Production Engineering, Taylor and Francis 1977:649–62.
[4] Khouja M, Michalewicz Z, Wilmot M. The use of genetic algorithms to solve the economic lot size scheduling problem. European Journal of Operational Research 1998;110(3): 509-24.
[5] Yang XS, Deb S. Cuckoo search via L´evy flights. Proc. World Congress on Nature & Biologically Inspired Computing 2009; 210-14.
[6] Dobson G. The Economic Lot Scheduling Problem: Achieving Feasibility using Time-Varying Lot Sizes. Operation Research 1987; 35(5): 764-71.
[7] Hanssmann F. Operations Research in Production and Inventory. John Wiley and Sons 1962;158-60.
[8] Kennedy J, Eberhard RC. Swarm intelligence. Morgan Kaufmann Publishers 2001.
[9] Aytug H, Khouja M, Vergara FE. Use of genetic algorithm to solve production and operations management problems: A review. International Journal of Production Research 2003;41(17): 3955–4009.
[10] Ben-daya M, Al-Fawzan M. A tabu search approach for the flow shop scheduling problem. European Journal of Operational Research 1998;109(1): 88–95.
[11] Eglese RW. Simulated annealing: A tool for operational research. European Journal of Operational Research 1990; 46(3): 271–81.
[12] Gaafar L. Applying genetic algorithms to dynamic lot sizing with batch ordering. Computers & Industrial Engineering 2006; 51(3):433–44.
[13] Press WH, Tehkolsky SA. Numerical Recipes 3rd Edition: The Art of Scientific Computing. Cambridge University Press, 2007.
[14] Sun H, Huang H, Jaruphongsa W. A genetic algorithm for the economic lot scheduling problem under extended basic period and power-of-two policy. CIRP Journal of Manufacturing Science and Technology 2009; 2(1): 29-34.
[15] Tasgetiren MF, Bulut O, and Fadiloglu MM. A discrete harmony search algorithm for the economic lot scheduling problem with power of two policy. IEEE World Congress on Computational Intelligence 2012.
[16] Raza SA, Akgunduz A. A comparative study of heuristic algorithms on Economic Lot Scheduling Problem. Computer & Industrial Engineering 2008; 55(1): 94-109.
[17] Gendreau M. An introduction to Tabu Search.Centre de recherche sur les transport 2002, 1-21.
[18] Adil SH, Ali SSA, Hussaan A, Raza K. Hybridization of Multiple Intelligent Schemes to Solve Economic Lot Scheduling Problem Using Basic Period Approach. *Life Sci J* 2013;10(2):2992-3005.

7/10/2013