1999年度日本オペレーションズ・リサーチ学会
春季研究発表会

ソフトウェア開発計画立案支援モデル

02004784 関西大学 *伊佐田百合子 ISADA Yuriko
01402374 関西大学 仲川勇二 NAKAGAWA Yuji

1．はじめに

コンピュータ，ネットワークの技術革新は目覚ましく，コンピュータシステムの社会活動における重要度は増加の一途をたどっており，システムの大規模化，複雑化，多様化が進んでいる．企業においても，めまぐるしい環境の変化に迅速に対応し，機動力を発揮するために，近年にも増して，企業経営を支える情報システムへの要求は大きい．通常，ソフトウェア開発プロジェクトは，1 年以上かかる物も珍しくないが，この様な経営の要求に応えるためには，できれば 3 ヶ月，遅くとも 6 ヶ月以内にシステムを構築し，効果を出さなければならない．それでなければ，システムが完成した時には，既に経営環境が変化してしまっている可能性があるからである．従って，ソフトウェア開発プロジェクトへの様々な要求（開発コスト，信頼性の確保等）を満たしつつ，それを管理し，納期までに開発を完了する事は非常に重要である．

ソフトウェアの開発工程の管理には様々なモデルが適用されるが，いずれの手法を取る場合でも，複数の作業項目が存在し，それらを複数のリソースで分担して開発にあたる．ソフトウェア開発プロジェクトを成功させるためには，適切な開発計画を立案し，その進捗状況を管理し，フォローしていく事がポイントとなる．

ソフトウェア開発計画の立案は，ターゲットとする納期内で如何に有効な要員計画を作るかが柱となる．しかし，複数ある作業のどれにどの要員を割り当ててもよいわけではなく，要員のスキルによって割り当て可能な作業とそうでない作業が存在し，割り当て可能な期間とそうでない期間が存在する（以降，リソース割り当て条件と呼ぶ）．これは，開発要員だけでなく，ソフトウェア開発に使用するリソース全般にいえる事である．更に，開発要員については，作業の平準化を図る事が重要である．これを怠れば，要員のモチベーションを下げ，円滑に開発作業が進まない等の考慮すべき点がある．一方，作業間にも，ある作業の前に必ず終了していなければならない作業があるなどの関連が存在する．

ソフトウェア開発の目的は，通常単一である事は少なく，費用をできるだけ安く，信頼性を高く，開発作業負荷を平準化し，できるだけ速く導入する等，複数の目的を要求される事が多い．しかし，これらの目的は，競合する事が多く，一般的に，全ての目的を同時に最大化(最小化)するような解は存在しない．ある目的の値を改善するためには，少なくとも他の 1 つの目的の値を改悪せざるを得ない．従って，複数の開発の目的を総合的に評価して満足できるソフトウェア開発計画を立案する必要がある．

2．代理目的を適用した多目的離散最適化問題

多目的離散最適化問題は次のように定式化される．

$$\begin{aligned} &\max && f(X) \\ &\text{s.t.} && g(X) \le b \end{aligned}$$

ここで，$X=(x_1,x_2,...x_n)$ は n 次元整数変数ベクトル，$f(X)=(f_1(X),f_2(X),...f_m(X))$ は m 次元ベクトル目的関数，$g(X)$ は制約関数である．

目的関数の最適値に ε だけの幅を持たせ，すべての目的関数がその範囲内に入る実行可能解を列挙させる問題は標的問題と呼ばれ[1]，多目的離散最適化問題に代理目的を適用した論文[2]では，これに代理乗数 u を導入して，複数個の目的関数を単一の目的関数に変換している．

$$\begin{aligned} &\text{target} && \mu f(X) \ge \mu f^{opt} - \varepsilon \\ &\text{s.t.} && g(X) \le b \\ &\text{但し} && \sum_{i=1}^{m} \mu_i = 1, \mu \ge 0 \end{aligned}$$

ここで，$f^{opt}=(f_1^{opt},f_2^{opt},...,f_m^{opt})$ は，それぞれの目的関数を単独で使用した時の目的関数の最適値ベクトル，ε は，各目的関数を統合した目的関数 $\mu_1 f_1+\mu_2 f_2+...+\mu_m f_m$ の最適値からの許容値である．目的関数空間における

実行可能領域$f(X)$を，許容値εの深さで代理目的の超平面により切断し，代理乗数μにより超平面が$f(X)$を切断する方向を変化させる事によって，意思決定者が任意のサイズと性質のパレート最適解集合を効率よく対話的に求める事ができる．我々の開発したプログラムでは，例えば，3目的，800変数，10項目案程度の問題を実用的な時間内に解く事ができる．

3．ソフトウェア開発計画問題

できるだけ短い納期，少ない開発総時間で信頼性の高いソフトウェアを決められた費用内で開発したい場合を考える．ソフトウェアの開発には，n個のリソース（単純化のため，ここでは人的リソースである要員について考え，以降，要員と呼ぶ）によって処理されるm個の作業が存在するとする．ソフトウェアの開発総時間T_iは，次のようにあらわされる．

$$T_i = \sum_{j=1}^{n} t_{ij}$$

ここで，t_{ij}は，要員jにおける作業iの開発時間である．各作業は通常先行後続関係を有しており，それぞれ関連し合う一連の作業パスが生成される．その作業パスのうち、最も長いパスが，該当ソフトウェアの開発期間であり，最長パス内の最終作業の完了をもって開発プロジェクトが終了となる．従って納期を最短にするには，関連のある一連の作業を作業群とし、それぞれの作業群内での作業時間の最小化を図ればよい．

ソフトウェア開発計画立案時点では，信頼性は，予想残存バグ数を尺度として測定する事ができる．過去の開発実績から，同程度の規模，複雑性を有したモジュールの計測結果を用いて，バグ保有数を推定する．システム全体の信頼性を高めるためには，バグ検出率を高め，残存バグ数を最小化する．各モジュールの残存バグ数を$R_i(x_i)$とした時，それぞれがすべて同等の重要度を持つわけではないので，モジュールの重要度による重み付けω_iを考慮し，残存バグ数の平均値の最小化を図る．

一方，総コストは次のように表される．

$$C_i = \sum_{j=1}^{n} c_{ij}$$

ここで，c_{ij}は，要員jにおけるi作業の開発費用である．

作業iの開発時間を$T_i(x_i)$，残存バグ数$R_i(x_i)$，開発コストを$C_i(x_i)$とおく．x_iは作業iにおいて実施する項目案の番号である．

予定された費用をbとした時，ソフトウェア開発計画問題は以下のように定式化する事ができる．

$$\min \quad f_1(x) = \sum_{i=1}^{m} T_i(x_i)$$

$$f_2(x) = \frac{1}{m}\sum_{i=1}^{m} \varpi_i R_i(x_i)$$

$$\text{s.t.} \quad g(X) = \sum_{i=1}^{m} C_i(x_i) \le b$$

$$x_i \in \{1,2,\ldots k_i\}$$

ここで，$X = (x_1, x_2, \ldots . x_m)$であり，$k_i$は作業iの項目案の個数である．ここで，各標的関数を単一で使用した時の問題として解き，実行可能解を得る．解の候補を限定するために，εを変化させ，多目的離散最適化アルゴリズムを用いて，標的問題を解き，標的解のサイズを変化させる．リソース割り当て条件を考慮し，割り当て不可能な解を削除するとともに，作業の先行関係を考慮して要員の作業に対する輻輳を取り除く．更に，ソフトウェア開発の目的の中での優先順位によってμを変化させ，εを操作する事で解集合のサイズを変化させて，総合的に目的に合った解を導き出す．このように，μ とεを操作して対話形式でパレート曲面の近傍を探索し，選好最適解を得る事で，複数の目的を総合的に満足するソフトウェア開発計画を立案する事が可能となる．

参考文献

[1] 疋田光伯, 仲川勇二, 伊藤俊英, 宮下文彬, 宮地功: 多目的離散最適化アルゴリズムの評価, 京都大学数理解析研究所考究録, No. 981, pp. 64-70(1997)

[2] 仲川勇二, 疋田光伯: 離散数理と連続数理における最適化理論, 京都大学数理解析研究所考究録, No. 1015, pp. 24-31(1997)