

直交表を利用した、データベースのシステム変数の最適化の検討

道明誠一

株式会社日立製作所システム開発研究所

システムの構成や利用条件に合わせてデータベースシステムなどの性能を引き出すには、システム変数を調整する必要がある。しかし、システム変数の組合せは膨大であり、その全てを網羅した測定結果から最良の物を選ぶのは不可能である。そこで、実務者が自らの経験を生かして、システム変数の選択並びに最適化を計ることが多い。しかし、データベース管理システム(DBMS)に関する知識も必要となるので、専門外の技術者にとって、調整作業が難しかった。本論文では、直交表の手法を応用し、システム変数の最適化を計る手法を提案する。主にハードウェアの機能評価に利用されていた直交表を、オープンソースソフトウェア(OSS)の性能評価について試行した。その結果、OSSの経験がなく、DBMSの知識に乏しい技術者であっても、直交表と回帰分析を併用した組合せテストを行うことで、適切なシステム変数を設計できる見通しを得た。

A startup DBMS parameter design using orthogonal arrays

Seiichi Domyo

Hitachi, Ltd., Systems Development Laboratory

It is difficult to optimize parameters of systems such as DBMSs. Usually, such work is done based on engineer's experience and know-how, which cannot be counted on ordinary users and nonspecialists. Being used many kinds of open source software under various conditions, the situations in which nonspecialists have to design the parameters are increasing. So help for optimizing such parameters for nonspecialists is desired. This paper proposes a method for tuning parameters by using orthogonal arrays, which is usually used hardware testing. An experiment of the method adapted to an open source DBMS shows that the method has good prospects.

1. はじめに

近年、オープンソースソフトウェア(OSS)は、オペレーティングシステムをはじめとし、実システムへの適用が普通となりつつある。そのうち、OSSのデータベース管理システム(DBMS)は、商用版のDBMSとくらべて小規模な運用に適しており、システムへの導入も容易であるので普及が進んでいる。しかし、OSSのDBMSの適用はあるが、一部のシステムを除くと、その詳細は不明である。そのため、運用時に、データベースの保守性に関する潜在的な問題を抱えている恐れがある。たとえば、利用者の増加に応

じて、この問題が顕在化すると、予定外の保守作業が発生する。また、データベース(DB)を保守しても、当初の性能要件を満せないシステムとなる最悪の場合もある。その場合に、アプリケーション(AP)の変更やシステムの破棄を余儀なくされる。

この問題を解決するには、DB設計やAP開発の各段階において、データベースの性能を予測することが望ましい[1]。しかし、DB設計において、将来の性能低下を予測し、その問題を解消するための手段は必ずしも確立していない。

他方、運用時に、DBの性能を引き出す手段の一つとして、DBMSの動作を制御する変数(以下、システム変数と記す)の変更が知られている[2]。

この変更手段を使用すると、DBMS層やOS層に関する性能上の不具合(ボトルネック)の一部について解消できる。システム変数で対処できるボトルネックの具体例として、a)一部のバッファやキャッシュの占有、b)特定の入出力処理の集中、c)多クライアント接続による競合、があげられる[2]-[4]。

本研究の目的は、DBの設計時において、DBMSのシステム変数の最適化により、運用時のDB性能低下を予防する手段の提供である。そのために、様々なシステム変数の組合せを変更し、ベンチマークソフトウェアによる性能予測を踏まえて、システム変数を調整する手段を検討した。

しかし、評価対象となるシステム変数の数は膨大である。そのために、(A)組合せの網羅率が向上するテストを利用する、(B)汎用的な基準により、変数を選択する、(C)個々の最適値ではなく、組合せとしての最適化を図る、という課題は残っている。

そこで、システム変数の最適化手法として、つぎの手順を提案する。

- (1) 直交表の手法[6]を採用し、システム変数の組合せテストの数を減らす
- (2) (1)の組合せテストの結果に対して、回帰分析[7]を使用し、システム変数を選択する
- (3) (2)のシステム変数について、最適値の組合せの効果について検証する

2. 関連研究

DBMSの機能として、自律化技術の搭載が望まれている。その一環として、たとえば、負荷変動や資源占有を監視し、システム変数を自動調整する技術の開発が進められている[5]。

このように、統計情報をもとに性能を予測し、システムソフトウェアの動作を制御する技術は、DBMSに限らない。関連研究として、たとえば、つぎの研究を紹介する。

- ・ オペレーティングシステムが、プロセッサに関する2変数をもとに、プロセス単位で動作周波

数を調整する[13]

- ・ ルータ装置が、TCPパケットに関する4変数をもとに、パケットを棄却する [14]

本研究が、関連研究と類似する点は、つぎのとおりである。

- ・ 回帰分析の手法を用いて、性能評価用のモデルを構築する
- ・ 性能を説明する変数同士は無相関である、という仮定にもとづいている

一方で、本研究は、関連研究と異なる点は、つぎのとおりである。

- ・ DBのシステム変数に回帰分析を適用する場合に、関連研究が扱う変数とくらべて圧倒的に多く、そのままではテストができない

この新たな問題を解決するのに、3以下で述べるように、直交表を利用した。

直交表の利用は、ハードウェアへの適用事例は存在する。しかし、ソフトウェアテスト、とくに、性能予測の分野では、調べた範囲では、みあたらなかった。たとえば、文献[16]は、ソフトウェアテストであるが、不良検出を目的とした機能評価である。

3. 直交表を利用した組合せテストの提案

まず、利用する直交表の概要について説明する。つぎに、システム変数に適用するための、検討課題を整理する。

3.1 直交表L36'の採用

直交表(orthogonal array)とは、どの配列を取っても同じ組合せが同じ数だけある(直交している)性質を備えた配列である。この性質を使うと、数少ないテスト数で最適な組合せを求めることができるので、ブラックボックステストの手法の一つとして使用されている[6]。

組合せテストに使う直交表には幾つかの種類があるが、その使い分けはつぎのとおりである[19]。

¹ 直交表の表記について述べると、たとえば、直交表L8とは、8行と4列(4変数が2つの値をとる)からなる配列であることを示している。

- ・ 直交表は、L8, L16, L27などの素数べき型直交表と、L12, L18, L36などの混合系直交表に分類できる
 - ・ 素数べき型直交表は因果関係の定量化に、混合系直交表は相関関係の定量化にそれぞれ向いている
 - ・ 混合系直交表L36は、L18とくらべて交互作用の影響をより受けにくい
- 他方、性能要件の要件は、つぎのとおりである。
- ・ 評価対象となるシステム変数の数が、20個を越えている
 - ・ 評価対象となるシステム変数の値が、2つあるいは3つの値をとりうる
 - ・ システム変数の最適値の組合せで性能低下が起こるのか、すなわち、交互作用が現れるのか不明である

直交表L36は、36行と23列からなる配列であり、上記の要件を満たしている。それゆえに、本提案では、混合系直交表L36を採用した。

使用した直交表L36の構成を示す(表1)。

ここで、行とは、テストの数に相当する。36行とは、36通りのテストである。列とは、変数の数に相当する。23列とは、最大23変数をとりうる。このうち、11変数(変数A～K)が2つの値、12変数(変数L～W)が3つの値をそれぞれとりうる。

したがって、直交表L36に23個のシステム変数を割り付けた組合せテストは、36通りで済む。それは、10億通り($2^{11} \times 3^{12}$)を越える、総当りの代表として、テストすることになる。

3.2 課題と解決のアプローチ

本研究では、1で述べた課題について、つぎのように解決した。

- 組合せの網羅率が向上するテストとして、直交表L36を利用する(その理由は、3.1と5.3で述べる)
- 性能予測並びに変数選択の汎用的な基準として、重回帰分析並びにAICを使用する(AICを使用した理由は5.2で述べる)
- 個々の最適値ではなく、組合せとしての最適化

を図るために、ベンチマークソフトウェアを用いて、検証した

3.3 提案手法

以下では、3.2で触れた課題(A)(B)(C)に関連して、直交表L36を利用する上での手順を示す。

- DBMSのマニュアル[8]と文献[2]-[4]を参考とし、二百を越えるシステム変数のうちで、とくに有用と思われるシステム変数を23個に絞り込む
- (1)の23個のシステム変数の構成に、偏りがないように配慮した。この変数の内訳は、a)一部のバッファやキャッシュの占有に関する変数7個、b)特定のディスク入出力処理の集中に関する変数8個、c)多クライアント接続による競合に関する変数8個である
- 36通りの組合せテストに対して、有意と認められたシステム変数のみを調整し、より詳細な条件下(本報告では、測定するクライアント数を増やし、3回繰り返して平均を取った)で検証する

4. 提案手法の評価

提案手法の実用性について、1回の組合せテストの結果で評価した。

DBMSの専門家であっても、システム変数の組合せテストは、ログ解析と変数調整の繰り返しをとまなう。DBの性能予測並びにシステム変数の最適化という手順を無制限に繰り返すと、試行錯誤に陥りやすい。しかし、直交表を用いることで、その繰り返しが、できれば1回、多くとも数回の試行で済めば、実用性は向上する、と考える。

4.1 評価環境と測定条件

4.1.1 評価環境

(1)測定環境

図1に、評価環境のシステムの構成を示す。システムの規模として、小規模(百人規模)の導入を想定した。その構成は、百台規模のクライアントマシンと、1台サーバ装置(APサーバとDBサーバと共用マシン)である。ただし、クライアント装置を用意せずに、サーバ装置単体で測定することにした。SysBench

は、サーバマシンと別装置で実行することもできるが、運用に合わせてMySQLと同じ装置で実行した。

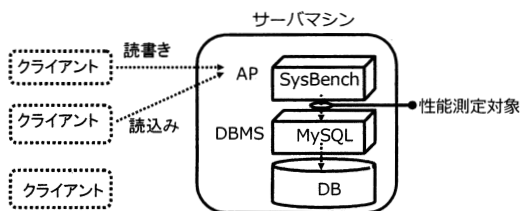


図1 測定環境のシステム構成

測定に使用したMySQL²は、2つのライセンスがあり、企業向けの商用版と開発者コミュニティ向けのOSS版に分かれる。今回は、OSS版を選択し、とくに、MySQL5.0という版のバイナリプログラムを使用した。

測定環境のサーバ装置(プロセッサ: Intel Pentium³D CPU 2.80GHz, 主記憶:4GB, HDD: ATA 80GB)にインストールしたMySQLサーバ[8]について、SysBenchベンチマーク[9]を用いて測定する。

(2)解析環境

解析環境のパソコンにインストールしたR言語(R本体[10]とR Commanderパッケージ[11])を使用し、重回帰分析を行う。

R言語による統計処理並びに操作手順は、文献[7]の内容に従った。

4.1.2 測定条件

SysBenchベンチマークを用いて生成する、評価用テーブルの規模は100万行で、同時接続するクライアント数は、最小1から最大64までとした。また、DBの更新性能あるいは参照性能はそれぞれ、ベンチマークの実行オプションをそのまま利用した。

以下にあげる3つの場合について、トランザクシ

²MySQLは、MySQL ABのアメリカ合衆国、欧州連合およびその他の国における登録商標です。

³ IntelおよびPentiumは、Intel Corporationのアメリカ合衆国及びその他の国における登録商標です。

ン性能を測定した。測定に際して、システム変数の組合せは、MySQLの推奨値(my-large.cnfという設定ファイルで指定する場合)の組合せと、最適化を施した組合せ(同推奨値の一部のパラメータを変更した)について、相対性能で比較した。

- (A) 更新性能
- (B) 参照性能
- (C) 参照・更新性能

4.2 測定結果

4.2.1 更新性能でのシステム変数の最適化

更新のシステム変数の最適値に対して、検証した(図2)。あわせて、その最適値を用いた場合の参照性能への影響も検証した。

測定結果はつぎのとおりである。

- (1) システム変数23個をテストした。そのうち4個が、更新性能と相関関係にあった
- (2) (1)の4変数を選択し、DBMSを調整し、検証した。そのときの更新性能は、MySQLの推奨値を利用するときの、最大+8%となった
- (3) 一方、参照性能は、-3%から+3%の間で変動した

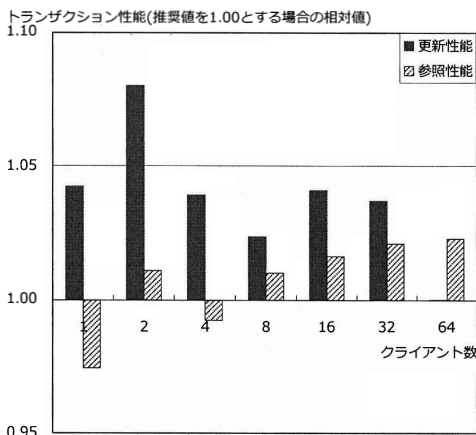


図2.更新性能とシステム変数の最適化

4.2.2 参照性能でのシステム変数の最適化

参照性能のシステム変数の最適値に対して、検証した(図3)。あわせて、その最適値を用いた場合の更新性能への影響も検証した。

測定結果はつぎのとおりである。

- (1) 4.2.1と同じシステム変数23個をテストした。そのうち7個が、参照性能と相関関係にあった
- (2) (1)の7変数を選択し、DBMSを調整し、検証した。そのときの参照性能は、MySQLの推奨値を利用するときの、最大+4%となった
- (3) 一方、更新性能は、最大+7%となった

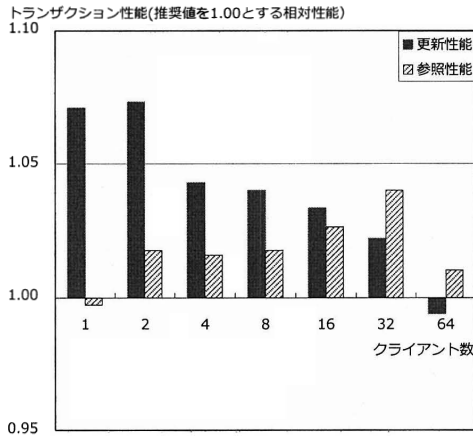


図 3.参照性能とシステム変数の最適化

4.2.3 参照更新性能でのシステム変数の最適化

参照性能と更新性能の両方について最適値について検証した(図4)。この測定では、更新性能は良好であったが、参照性能において、若干の性能低下が見られた。これは、回帰モデルの構築が更新性能向けとなり、過調整となったと推定する。

測定結果はつぎのとおりである。

- (1) 4.2.1と4.2.2の各36通りの組合せテストを合わせた計72通りの測定結果について回帰分析を行う。その結果、システム変数23個のうちの5個が、更新性能と参照性能の何れにも相関関係にあった
- (2) (1)の5変数を選択し、DBMSを調整し、検証した。そのときの更新性能は、MySQLの推奨値を利用するときにくらべて、最大+7%となった
- (3) 参照性能は、-2%から+3%の間で変動した

5. 提案方式の評価

5.1 直交表と制約条件

本報告では、変数に関する制約要件を考慮していない。そこで、測定後、参考文献[2]-[4]を参照し、該当システム変数の設定の妥当性について確認した。本報告の範囲では、とくに禁則に触れた設定は見当たらなかった。

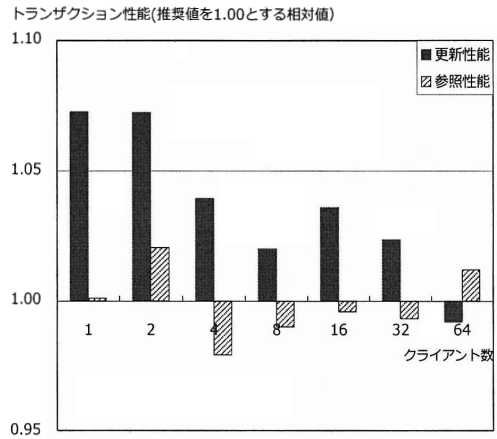


図 4.参照更新性能とシステム変数の最適化

5.2 直交表と回帰分析

回帰分析では、システム変数の選定基準として、AIC[18]を利用した。

この手順に関連し、ソフトウェアテストについての基礎研究がある。文献[15]によると、直交表とAICとの連携は、必ずしも相関関係の精度が良いとはいえない。しかし、本報告の範囲では、実用上、とくに問題とはならなかった。

また、本研究は、DBの自律制御という研究分野[5]からみると、実システムでの適応性や制御のリアルタイム性に関する検討は足りない。ただし、本研究の目的は、運用時の性能保証ではなく、設計時の性能低下の予防である。この目的において、提案方式は、無制限な調整が発生しにくいので、十分な実用性を備えていると考える。

5.3 直交表と全ペアアルゴリズム

組合せテストのテスト数を減らす、別の手法として、全ペアアルゴリズムがある[6]。全ペアアルゴリズムの一つとして、PICT[12]について検討した。23個の変数を扱うときに、PICTには、a)直交表L36(36通り)とくらべてよりテスト数を減らせる(20通り)、b)禁則条件を記述できる、という特長がある。その一方で、直交表L36は、PICTにくらべてテスト数が多いが、その分、2変数のみならず、3変数以上の組合せについて網羅している可能性が高い。

直交表L36に対して、全ペアアルゴリズムPICTの組合せテストの結果を示す。同じ23個の変数を使ってそれぞれ組合せを作り、クライアント数1,8,64でそれぞれ1回ずつ測定した。

図5は、4.2.1の更新性能に関する両者の比較である。図のグラフにおいて、点はトランザクション性能の相対値を記している。クライアント数の3点を結ぶ線は、ベンチマークとして連続して計測したことを示している。

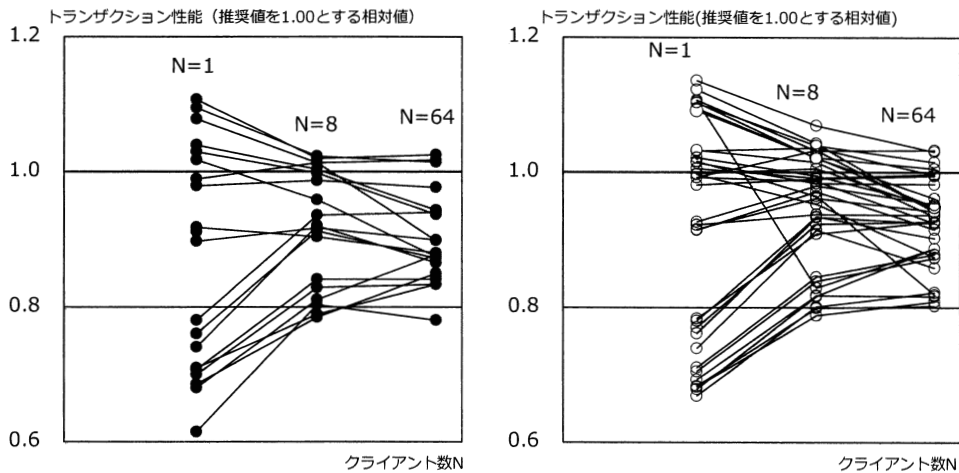
両者には大きな違いは見られない。本報告では、網羅率がより良いとされている直交表L36を選んだが、全ペアアルゴリズムPICTでも同様に、組合せテストの絞り込みができると予想する。

6. おわりに

データベースのシステム変数の最適化に関する、提案方式の実用性を検証した。また、代表的なオープンソースソフトウェアであるMySQLと、ベンチマークSysBench、並びに解析ソフトウェアR言語を用いて、その実現性を検証した。

検証した内容は、以下のとおりである。

- (1) 直交表の手法を応用した。総当り10億を越える組合せテストの対象を36通りに絞り込んだ
- (2) 回帰分析を利用した。評価用の線形回帰モデルを構築し、変数を選択する。さらに最適値を求めた



(a) 全ペアアルゴリズムPICTによる組合せテスト (b)直交表L36による組合せテスト

図 5. 直交表と全ペアアルゴリズムの比較

7. 参考文献

- [1] 後藤, 名和, 五嶋, 井原著: SE のための Oracle チューニングハンドブック, ソフトバンクパブリッシング(2003)
- [2] Schwartz, Zaitsev, Tkachenko, Zawodny, Lentz, Balling 著: High Performance MySQL 2nd Edition (6. Optimizing Server Settings pp.265-304), O'reilly & Associates Inc(2008)
- [3] 松信著: 現場で使える MySQL(13 章-14 章 パフォーマンスチューニング pp.280-329), 翔泳社(2006)
- [4] Pachev 著, 伊藤・田中・吉川監訳, 菅野訳: 詳解 MySQL(5 章 設定変数 pp.79-106), オライリー・ジャパン(2007)
- [5] Lightstone, Schiefer, Zilio, Kleewein. Autonomic Computing for Relational Databases: The Ten Year Vision. In Proceedings of Workshop on Autonomic Computing Principles and Architectures (AU-COPA2003), August 2003
- [6] コーブランド著, 宗訳: はじめて学ぶソフトウェア技法(第 6 章ベア構成テスト pp61-90), 日経 BP(2005)
- [7] 荒木著: R と R コマンダーではじめる多変量解析(3.7 説明変数に質的変数を含む回帰分析 pp.91-99), 日科技連(2007)
- [8] MySQL Community Server <http://dev.mysql.com/>
- [9] SysBench: a system performance benchmark <http://sysbench.sourceforge.net/>
- [10] R 2.7.1 Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>
- [11] John Fox, with contributions from Michael Ash, Theophilus Boye, Stefano Calza, Andy Chang, Philippe Grosjean, Richard Heiberger, G.Jay Kerns, Renaud Lancelot, Matthieu Lesnoff, Samir Messad, Martin Maechler, Duncan Murdoch, Erich Neuwirth, Dan Putler, Brian Ripley, Miroslav Ristic and and Peter Wolf. (2008). Rcmdr: R Commander. R package version 1.4-5. <http://www.r-project.org>, <http://socserv.socsci.mcmaster.ca/~jfox/Misc/Rcmdr>
- [12] PICT: Pairwise Independent Combinatorial Testing tool <http://www.pairwise.org/tools.asp>
- [13] 金井, 佐々木, 近藤, 中村, 並木, 統計情報に基づく省電力 Linux スケジューラ pp.9-16, 2007-OS-106-(2), 情報処理研究報告 Vol.2007 No.83
- [14] 江口, 大崎, 村田, 多変量解析を用いたアクティブキュー管理機構の性能評価に関する検討, pp.37-40, NS2002-25, 信学技報 Vol.102 No.20
- [15] 小谷, 須田, 汎用的なソフトウェア自動チューニング機構のための実験計画法の応用の検討, pp.193-198, 2006-HPC-107-(33), 情報処理研究報告 Vol.2006, No.87
- [16] 山本, 秋山, 直交表を利用したソフトウェアテスト-HAYST 法-, JaSST'04: Japan Symposium on Software Testing 2004
- [17] 宮川著: 品質を獲得する技術(5.5 L36 直交表の性質 pp.168-173), 日科技連(2000)

表1. 直交表 L36

| # | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 5 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| 7 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 2 | 3 | 1 | 2 | 3 | 3 | 1 | 2 | 2 | 3 |
| 8 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 1 | 2 | 3 | 1 | 1 | 2 | 3 | 3 | 1 |
| 9 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 1 | 2 | 3 | 1 | 2 | 2 | 3 | 1 | 1 | 2 |
| 10 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 1 | 1 | 3 | 2 | 1 | 3 | 2 | 3 | 2 | 1 | 3 | 2 |
| 11 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 3 | 2 | 1 | 3 | 1 | 3 | 2 | 1 | 3 |
| 12 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 3 | 2 | 1 | 3 | 2 | 1 | 2 | 1 | 3 | 2 | 1 |
| 13 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 2 | 3 | 1 | 3 | 2 | 1 | 3 | 3 | 2 | 1 | 2 |
| 14 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 1 | 3 | 2 | 1 | 1 | 3 | 2 | 3 |
| 15 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 3 | 1 | 2 | 3 | 2 | 1 | 3 | 2 | 2 | 1 | 3 | 1 |
| 16 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 3 | 2 | 1 | 1 | 3 | 2 | 3 | 3 | 2 | 1 |
| 17 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 3 | 1 | 3 | 2 | 2 | 1 | 3 | 1 | 1 | 3 | 2 |
| 18 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 3 | 1 | 2 | 1 | 3 | 3 | 2 | 1 | 2 | 2 | 1 | 3 |
| 19 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 1 | 3 | 3 | 3 | 1 | 2 | 2 | 1 | 2 | 3 |
| 20 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 3 | 2 | 1 | 1 | 1 | 2 | 3 | 3 | 2 | 3 | 1 |
| 21 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 3 | 1 | 3 | 2 | 2 | 2 | 3 | 1 | 1 | 3 | 1 | 2 |
| 22 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 3 | 3 | 1 | 2 | 1 | 1 | 3 | 3 | 2 |
| 23 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 1 | 1 | 2 | 3 | 2 | 2 | 1 | 1 | 3 |
| 24 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 2 | 3 | 1 | 3 | 3 | 2 | 2 | 1 |
| 25 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 1 | 3 | 2 | 1 | 2 | 3 | 3 | 1 | 3 | 1 | 2 | 2 |
| 26 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 1 | 3 | 2 | 3 | 1 | 1 | 2 | 1 | 2 | 3 | 3 |
| 27 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 3 | 2 | 1 | 3 | 1 | 2 | 2 | 3 | 2 | 3 | 1 | 1 |
| 28 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 3 | 2 | 2 | 2 | 1 | 1 | 3 | 2 | 3 | 1 | 3 |
| 29 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 3 | 3 | 3 | 2 | 2 | 1 | 3 | 1 | 2 | 1 |
| 30 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 3 | 2 | 1 | 1 | 1 | 3 | 3 | 2 | 1 | 2 | 3 | 2 |
| 31 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 1 | 2 | 1 | 1 |
| 32 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 3 | 1 | 3 | 3 | 2 | 3 | 2 | 2 |
| 33 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 3 | 2 | 2 | 2 | 1 | 2 | 1 | 1 | 3 | 1 | 3 | 3 |
| 34 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 1 | 3 | 1 | 2 | 3 | 2 | 3 | 1 | 2 | 2 | 3 | 1 |
| 35 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 3 | 1 | 3 | 1 | 2 | 3 | 3 | 1 | 2 |
| 36 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 3 | 2 | 3 | 1 | 2 | 1 | 2 | 3 | 1 | 1 | 2 | 3 |